

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCUOLA DI INGEGNERIA E ARCHITETTURA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

DIPARTIMENTO DI INFORMATICA – SCIENZA E INGEGNERIA

Tesi di Laurea in Reti di Calcolatori T

**Ambienti di servizi per applicazioni in
Mixed Reality per MS HoloLens**

CANDIDATO

Robin Esposito

RELATORE

Chiar.mo Prof. Ing. Antonio Corradi

CORRELATORI

Prof. Ing. Luca Foschini

Ing. Alessandro Rizzoli

Anno Accademico 2015/2016

Sessione II

RINGRAZIAMENTI

Desidero ringraziare i professori Antonio Corradi e Luca Foschini, che mi hanno guidato nella realizzazione di questa tesi di laurea e Alessandro Rizzoli, fondatore di GetConnected, che con grande disponibilità mi ha fornito l'opportunità di lavorare ad un progetto innovativo ed interessante.

INDICE

| | |
|---|-----------|
| INTRODUZIONE | 7 |
| 1. COMPUTER-MEDIATED REALITY: REALTÀ VIRTUALE, AUMENTATA O MISTA | 9 |
| 1.1. <i>Computer-mediated Reality</i> | 9 |
| 1.2. <i>Reality-virtuality continuum</i> | 10 |
| 1.3. <i>Virtual Reality (VR)</i> | 11 |
| 1.4. <i>Augmented Reality (AR)</i> | 13 |
| 1.5. <i>Mixed Reality (MR)</i> | 15 |
| 2. VISORI E TECNOLOGIE | 19 |
| 2.1. <i>Visori per la realtà virtuale</i> | 19 |
| 2.1.1. Oculus Rift | 19 |
| 2.1.2. HTC Vive | 20 |
| 2.1.3. Google Cardboard | 21 |
| 2.1.4. Samsung Gear VR | 22 |
| 2.2. <i>Visori per la realtà aumentata</i> | 23 |
| 2.2.1. Google Glass | 23 |
| 2.3. <i>Visori per la realtà mista</i> | 25 |
| 2.3.1. Microsoft HoloLens | 25 |
| 2.3.2. Meta Development Kit 2 | 26 |
| 3. MICROSOFT HOLOLENS: SPECIFICHE TECNICHE, STRUMENTI DI SVILUPPO E FUNZIONALITÀ | 29 |
| 3.1. <i>Caratteristiche tecniche e di sviluppo</i> | 29 |
| 3.1.1. Caratteristiche hardware | 29 |
| 3.1.2. Strumenti di sviluppo e requisiti | 30 |
| 3.1.3. Piattaforma di esecuzione e sistema operativo | 31 |
| 3.2. <i>Funzionalità ed interazione</i> | 32 |
| 3.2.1. Tipologie di input: sguardo, gesti e comandi vocali | 32 |
| 3.2.2. Spatial Mapping | 33 |
| 3.2.3. Spatial Sound | 34 |
| 3.2.4. Condivisione degli ologrammi | 35 |
| 3.3. <i>Stato attuale dello sviluppo per MS HoloLens</i> | 36 |
| 4. APPLICAZIONE: PREMESSE E OBIETTIVI | 39 |
| 4.1. <i>Introduzione e premesse allo sviluppo dell'applicazione</i> | 39 |
| 4.1.1. Campo di interesse dell'applicazione | 39 |
| 4.1.2. Requisiti del cliente | 40 |
| 4.1.3. Obiettivi dell'applicazione | 41 |
| 4.2. <i>Strumenti di sviluppo</i> | 42 |
| 4.2.1. Unity, piattaforma di sviluppo tridimensionale | 43 |
| 4.2.1.1. Costruzione della scena di un'applicazione | 43 |
| 4.2.1.2. Integrazione del codice nella scena dell'applicazione | 45 |
| 4.2.2. Blender, software per la manipolazione di modelli 3D | 45 |
| 4.2.3. HoloToolkit, collezione di componenti per Windows Holographic | 47 |
| 5. ARCHITETTURA E IMPLEMENTAZIONE DELL'APPLICAZIONE | 49 |
| 5.1. <i>Struttura dell'applicazione</i> | 49 |
| 5.1.1. Casi d'uso | 49 |
| 5.1.1.1. Descrizione dei casi d'uso | 50 |
| 5.1.2. Organizzazione della scena tridimensionale | 52 |

| | |
|--|-----------|
| 5.1.3. Set di comandi per l'interazione | 54 |
| 5.1.3.1. Base Commands..... | 55 |
| 5.1.3.2. Rotation Commands..... | 56 |
| 5.1.3.3. Sofa Choice e Color Choice..... | 57 |
| 5.2. Gestione dei dati di input..... | 58 |
| 5.2.1. Managers..... | 58 |
| 5.2.2. Gaze Manager | 59 |
| 5.2.3. Gesture Manager | 61 |
| 5.2.4. Speech Manager | 62 |
| 5.3. Operazioni di interazione..... | 63 |
| 5.3.1. Memorizzazione dello stato attuale..... | 63 |
| 5.3.2. Selezione del modello ed avvio dell'interazione | 64 |
| 5.3.3. Posizionamento del modello nello spazio reale | 65 |
| 5.3.4. Rotazione del modello | 66 |
| 5.3.5. Modifica del colore del modello | 67 |
| 5.3.6. Rimozione del modello dallo spazio | 69 |
| 5.3.7. Comandi vocali | 70 |
| 6. RISULTATI OTTENUTI E PERFORMANCE | 71 |
| 6.1. Flusso di esecuzione dell'applicazione | 71 |
| 6.2. Test di utilizzo | 72 |
| 6.3. Costi di sviluppo dell'applicazione | 74 |
| 6.4. Presentazione dell'applicazione a Farete 2016..... | 75 |
| CONCLUSIONI E POSSIBILI SVILUPPI | 79 |
| BIBLIOGRAFIA | 81 |

INTRODUZIONE

Osservando l'evoluzione dei sistemi informativi negli ultimi anni, si nota la ricerca di una sempre maggiore integrazione degli elementi tecnologici nella vita quotidiana; l'utilizzo costante degli smartphone, i tablet che sostituiscono i supporti cartacei, tutti i dispositivi elettronici con cui veniamo in contatto ogni giorno ne sono un esempio concreto. L'interazione tra il mondo reale ed i dispositivi informatici suscita grande interesse: non solo consente un'elevata personalizzazione dell'esperienza dell'utente, ma apre la strada ad infinite possibilità di applicazione. Il culmine di questa commistione tra esperienza reale e dati virtuali si è raggiunto nel campo della Computer-mediated Reality, termine che indica l'insieme di tecnologie che offrono all'utente la percezione sensoriale di una realtà alterata tramite un'unità di elaborazione.

La Computer-mediated Reality è un campo piuttosto ampio, a cui appartengono varie tecnologie; per il presente lavoro di tesi l'attenzione è stata focalizzata sulla realtà mista, che si basa sull'inserimento di elementi virtuali, definiti ologrammi, nello spazio reale tramite appositi visori. La possibilità di creare un ambiente in cui oggetti concreti ed immagini computerizzate coesistono ed interagiscono tra loro può costituire il punto di partenza per le applicazioni più differenti. Gli utilizzi che sono stati finora ipotizzati e verso cui sono orientati correntemente gli sviluppi spaziano dall'intrattenimento, al training, al turismo, all'addestramento militare.

La realtà mista è una tecnologia ancora in fase di sviluppo, ma per cui si prevede una rapida diffusione: solo a marzo del 2016 è stato rilasciato dalla Microsoft il visore HoloLens, specifico per la Mixed Reality. Per questo lavoro di tesi mi è stata offerta l'opportunità di lavorare con GetConnected, un'azienda bolognese di sviluppo software e una tra le prime in Italia ad ottenere di poter lavorare con l'HoloLens. In particolare, collaborando con Poltronesofà, è stata progettata e realizzata un'applicazione che consentisse di applicare l'idea della realtà mista al design d'interni: gli utenti, forniti di apposito visore, potranno utilizzare l'applicazione per inserire nelle loro case l'ologramma di un divano, selezionandolo da un apposito catalogo. Potranno poi muoverlo e ruotarlo nello spazio per valutarne l'effetto rispetto all'ambiente, come se si trattasse di un reale elemento di arredamento; grazie alla virtualizzazione, però, potranno usufruire di vantaggi come poter

accedere liberamente a tutti i modelli di divani disponibili e poterne modificare in ogni momento le caratteristiche.

Nella presente tesi, dopo una prima parte dedicata all'analisi del concetto di realtà mediata e delle sue varie specializzazioni, verranno illustrati i principali dispositivi hardware presenti attualmente sul mercato, analizzandone le caratteristiche, gli aspetti innovativi e la diffusione raggiunta al momento. Si passerà quindi ad esaminare nel dettaglio il visore Microsoft HoloLens, impiegato per la realizzazione del progetto di laurea, per arrivare infine alla descrizione dello sviluppo dell'applicazione.

In particolare, la tesi sarà così strutturata: nel primo capitolo verranno analizzati il concetto di Computer-Mediated Reality e le sue sottocategorie, ovvero Virtual, Augmented e Mixed Reality. Nel secondo verranno illustrati alcuni visori ritenuti particolarmente notevoli dal punto di vista dell'innovazione, suddivisi per campo di appartenenza; nel terzo ci si focalizzerà invece sull'HoloLens, descrivendone caratteristiche tecniche, specifiche di sviluppo, modalità di utilizzo ed eventuali limitazioni. Nel quarto capitolo saranno descritte le premesse, i requisiti di sviluppo e gli obiettivi relativi all'applicazione sviluppata, mentre nel quinto ne verrà illustrata l'effettiva struttura ed implementazione. Nel sesto capitolo, infine, verranno descritti il flusso di esecuzione dell'applicazione, i test effettuati ed il feedback ricevuto dagli utilizzatori.

1. COMPUTER-MEDIATED REALITY: REALTÀ VIRTUALE, AUMENTATA O MISTA

1.1. *Computer-mediated Reality*

Il concetto di “realtà mediata” si riferisce all’alterazione della percezione sensoriale della realtà circostante, accessibile all’utente da un computer o tramite un dispositivo portatile o indossabile. I primi sviluppi in tale ambito risalgono al 1970: nello specifico, le ricerche erano orientate all’utilizzo di visori per la realtà mediata come supporto per non vedenti e ipovedenti. In quell’anno infatti Steve Mann, ricercatore ed inventore canadese, introdusse la prima generazione dei Digital Eye Glasses, inizialmente un elmetto indossabile collegato ad ingombranti computer e videocamere, ma destinato a evolversi nel tempo in compatti occhiali simili ai più noti Google Glass. Da allora, l’interesse per la realtà mediata è in costante aumento ed i campi di applicazione si sono notevolmente allargati.

Specialmente negli ultimi anni, gli investimenti hanno registrato una rapida impennata: dal luglio del 2015 al giugno del 2016 è stato investito un totale di 2 miliardi di dollari [Roe16]. Particolare interesse ha suscitato il campo dell’intrattenimento o, più esattamente, del gaming. Sono stati infatti distribuiti numerosi videogiochi ed appositi supporti hardware che utilizzano la realtà mediata per offrire un’esperienza di gioco più coinvolgente e realistica.

Ma non mancano altri campi di applicazione, che si delineano man mano che le tecnologie evolvono. Ad esempio, intorno al 2014 ebbero una forte diffusione le pubblicità che facevano uso della realtà mediata. Gli spot risultavano particolarmente efficaci poiché coinvolgevano l’utente richiedendo una sua partecipazione diretta. Possibili applicazioni che stanno attualmente suscitando l’interesse pubblico includono il turismo e la partecipazione ad eventi da remoto, l’arredamento di interni e l’istruzione, ma continuano a presentarsi nuove possibilità di utilizzo [Dim16].

La definizione stessa di realtà mediata è stata migliorata ed arricchita nel tempo e sono state definite tre sottocategorie, ognuna una particolare specializzazione del concetto iniziale: realtà virtuale (Virtual Reality o VR), realtà aumentata (Augmented Reality o AR) e realtà mista (Mixed Reality o MR). La differenza tra i diversi campi può essere labile, poiché va considerato che si tratta di tecnologie in costante evoluzione, i cui confini non sono ancora

perfettamente delineati. Per mettere ordine tra le varie categorie sarà utile definire un criterio di misura: il continuo reale-virtuale.

1.2. *Reality-virtuality continuum*

Il continuo reale-virtuale, introdotto da Paul Milgram nel 1994, rappresenta una scala di valori attraverso la quale confrontare le varie tecnologie appartenenti al campo della realtà mediata. Tale scala, in Figura 1, consente di quantizzare in che misura la realtà presentata all'utente è costituita da elementi virtuali e quanto invece è parte della realtà effettiva: i due estremi della scala sono infatti la virtual reality, integralmente fittizia, ed il mondo reale [Mil94].

Per la realtà virtuale, quindi, la classificazione è molto chiara: la realtà effettiva viene completamente nascosta all'utente e sostituita da un mondo fittizio generato tramite computer; ci troviamo quindi all'estremo "virtuale" del continuo. Nei casi della realtà aumentata e della realtà mista, invece, il confine è molto più labile e le differenze più sottili.

Augmented Reality e Mixed Reality verranno analizzati nel dettaglio nei paragrafi successivi; per ora ci limitiamo a darne una definizione generica ed a esaminarne le differenze. Il concetto di realtà aumentata consiste nella sovrapposizione al mondo reale di informazioni che possono essere utili all'utente: tali informazioni sono quindi aggiunte alla realtà effettiva, ma generalmente non sono rappresentazioni di oggetti realistici. Ad esempio, un'applicazione che, quando viene inquadrato un luogo di interesse turistico, mostri all'utente informazioni come prezzi, orari e quant'altro, rientra nel campo della realtà aumentata.

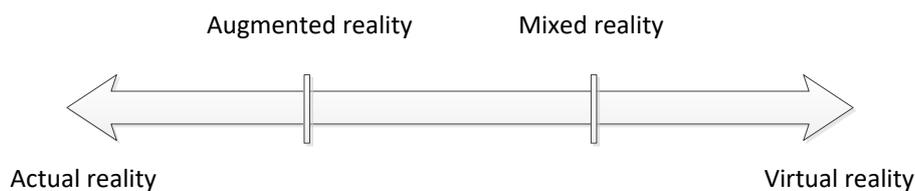


Figura 1 - Scala del continuo reale-virtuale

Sulla scala del continuo reale-virtuale ci troviamo quindi più vicini all'estremità della realtà, anche se non al punto estremo: l'utente continua ad avere percezione del mondo reale con l'aggiunta di alcune informazioni virtuali sovrapposte ad esso. Infine, nel caso della realtà mista, ci troviamo ad un punto intermedio tra la realtà virtuale e quella aumentata, e quindi anche ad un punto intermedio sulla scala del continuo: con la realtà mista si mira a raggiungere la massima integrazione possibile tra mondo reale e mondo virtuale. Come nella realtà aumentata, l'utente ha la percezione del mondo reale, al quale vengono aggiunti elementi virtuali; in questo caso, però, non si tratta di oggetti distaccati semplicemente sovrapposti, ma di rappresentazioni realistiche che interagiscono con la realtà.

In questo senso, la realtà mista assomiglia a quella virtuale, poiché ricostruisce spazi ed oggetti fisici dandone una percezione realistica all'utente, caratteristica che manca completamente nella realtà aumentata. Nella realtà mista gli oggetti virtuali vengono ancorati allo spazio fisico reale e non semplicemente sovrapposti: in questo modo l'utente ha la percezione di una realtà che è commista di elementi reali e virtuali. Un esempio potrebbe essere una scrivania a cui vengono aggiunti cassetti, ante o moduli attraverso un visore per la realtà mista: gli elementi virtuali saranno quindi ancorati al mobile reale e l'utente avrà percezione sia del mobile reale che degli elementi virtuali ad esso aggiunti.

1.3. Virtual Reality (VR)

Con "Virtual Reality" si intende una realtà simulata, più o meno realistica, che viene elaborata da un computer e presentata all'utente, generalmente tramite monitor oppure visori o altri dispositivi indossabili. La percezione della realtà virtuale può essere esclusivamente visiva o può essere accompagnata da altre percezioni sensoriali: spesso quelle uditive, ma talvolta anche tattili e olfattive.

Il termine "Virtual Reality" è stato reso popolare nel 1980 da Jaron Lanier, uno dei pionieri del campo e fondatore della società "VPL Research", a cui si deve lo sviluppo di alcuni dei primi dispositivi per la VR, come il "Data Glove" o lo "Eye Phone", che ebbero però scarsa diffusione. Negli anni successivi la popolarità della realtà virtuale crebbe rapidamente, tanto da diventare soggetto di numerose opere di narrativa, cinematografiche o anche artistiche.

Da allora le applicazioni della realtà virtuale si sono notevolmente ampliate. Senza dubbio uno dei campi in cui sono stati fatti i maggiori investimenti è quello del gaming, raggiungendo risultati notevoli in tempi particolarmente brevi per soddisfare le richieste del mercato, ma le possibili applicazioni che si stanno delineando sono svariate. Una delle tante è quelle dell'addestramento: la realtà virtuale viene utilizzata per simulare situazioni e scenari che sarebbe difficile realizzare praticamente, magari perché particolarmente costoso o pericoloso; ad esempio può essere impiegata per un addestramento militare, una simulazione di volo o come esercizio per un'operazione chirurgica.

Altri campi dell'intrattenimento in cui la realtà virtuale inizia ad essere ampiamente utilizzata sono la cinematografia e il turismo. Sono già piuttosto comuni video girati con telecamere a 360 gradi, fruibili sia tramite visore che tramite smartphone: su YouTube, ad esempio, dal marzo 2015 è possibile caricare e guardare filmati girati a 360 gradi. Inoltre, ad aprile del 2016 è stata integrata la possibilità di utilizzare lo "spatial audio", ovvero una riproduzione realistica dell'ambiente sonoro, che varia durante la riproduzione del video a seconda della posizione dell'utente, con l'obiettivo di rendere l'esperienza più veritiera possibile [You16]. Se questo tipo di tecnologia faccia propriamente parte della Virtual Reality è discutibile: è opinione diffusa che i video in 360 gradi rappresentino una nuova tipologia di media, sicuramente più interattiva di quelle tradizionali ma lontana dal concetto originario di realtà virtuale. Essendo tecnologie particolarmente nuove e in rapida evoluzione, risulta in ogni caso difficile fornirne una categorizzazione esatta, ma sono state incluse in questo elenco per dare una panoramica completa delle varie opzioni disponibili sul mercato [Pop16].



Figura 2 - Esempio di utilizzo della VR nel campo del turismo a Casa Batllò, Barcellona

Nel turismo, invece, la VR viene può essere impiegata per offrire agli utenti l'esperienza di un viaggio da remoto, ovvero l'esplorazione di un posto lontano senza doversi muovere realmente, ma immergendosi nella realtà virtuale attraverso un visore. Un altro utilizzo fatto della VR nello stesso campo è la ricostruzione di luoghi di interesse storico, o anche la visualizzazione di possibili strutture future non ancora realizzate al momento della visita. Un esempio di questo tipo di impiego della realtà virtuale è l'installazione presso Casa Batllò, a Barcellona, dove i visitatori vengono dotati all'ingresso di uno smartphone; avviando l'apposita applicazione in punti predefiniti dell'edificio, viene visualizzata una ricostruzione degli ambienti completi degli arredi originali, con la quale gli utenti interagiscono muovendo lo smartphone per esplorare gli spazi circostanti (Figura 2, tratta da <https://www.casabatllo.es/en/visit/videoguide>). Si tratta in questo caso propriamente di Virtual Reality poiché gli arredi non sono sovrapposti agli spazi reali, ma viene proposta una ricostruzione interamente virtuale degli ambienti.

1.4. Augmented Reality (AR)

L'Augmented Reality, o realtà aumentata, consiste nella sovrapposizione al mondo reale di informazioni come oggetti, immagini o testo generati in maniera virtuale. In questo caso non

abbiamo quindi una sostituzione della realtà effettiva, ma piuttosto un'alterazione della stessa. Il mondo reale viene generalmente ripreso da una telecamera, che analizza le immagini ricevute e le rielabora aggiungendo le informazioni necessarie, per poi ripresentarle all'utente.

Gli oggetti virtuali aggiunti dal computer sono generalmente poco realistici: non vengono incorporati nell'ambiente circostante, ma piuttosto sovrainpressi. Ovviamente non sono del tutto scollegati dalla realtà ma esiste sempre un punto di contatto rilevato dalla telecamera grazie al quale gli oggetti possono essere posizionati correttamente e con le giuste angolazioni. L'interazione che l'utente può avere con gli elementi virtuali è minima, quasi sempre limitata all'osservazione dei dati presentati.

Anche in questo caso esistono numerose possibili applicazioni: dal gaming, all'istruzione, al design, al turismo. Esistono vari tipi di supporti hardware che permettono di usufruire della realtà aumentata: le tipologie più comuni sono quella dei dispositivi head-mounted, ovvero dispositivi indossabili sulla testa, a forma di elmetto o di visore, e quella degli eyeglasses, dal design simile ad occhiali da vista. Un tipo di supporto non ancora commercializzato, in fase di sviluppo, sono delle lenti a contatto bioniche che ridurrebbero al minimo l'ingombro e il fastidio dei classici visori pur mantenendone le funzionalità.

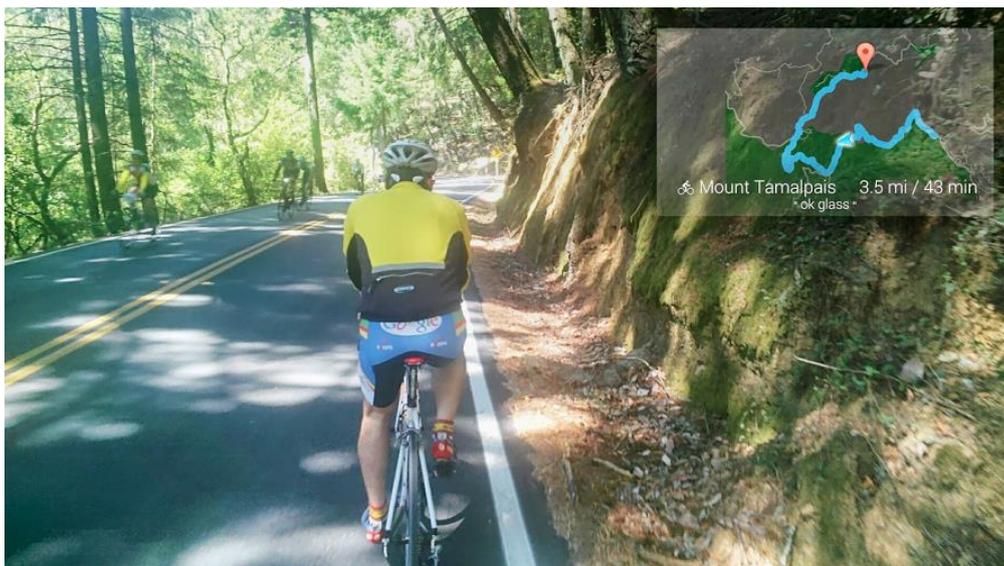


Figura 3 - Esempio di AR: indicazioni stradali visualizzate tramite Google Glass

Un utilizzo alternativo della AR, ormai piuttosto comune, è l'impiego in applicazione per smartphone: sfruttando le tecnologie già presenti in gran parte degli smartphone più diffusi, gli utenti possono usufruire della realtà aumentata senza ulteriori spese per appositi visori. Quasi tutti gli smartphone oggi in commercio dispongono infatti dei requisiti minimi per il supporto della realtà aumentata: una fotocamera per l'acquisizione delle immagini, localizzatori GPS e connessione wireless per il posizionamento, oscilloscopio e accelerometro per l'elaborazione dei movimenti dell'utente (Figura 3, tratta da <https://support.google.com/glass/answer/3405215?hl=en>). Questo tipo di utilizzo è stato sfruttato ampiamente nel campo del marketing: attraverso la AR l'utente può inquadrare il prodotto desiderato per ottenere ulteriori informazioni. Ad esempio, Ikea ha realizzato nel 2013 un'applicazione che rappresenta un'estensione del classico catalogo cartaceo: gli utenti sono invitati a posizionare il catalogo concreto in un punto della stanza per poi inquadrarlo con un qualunque smartphone. L'applicazione rileva l'immagine del catalogo all'interno dello spazio e, utilizzandolo come punto di riferimento, mostra una riproduzione virtuale dell'elemento di arredo scelto dall'utente [Sti13].

Anche nel campo del gaming è sempre più comune che vengano sfruttati dei comuni smartphone per la realizzazione della realtà aumentata; la qualità complessiva dell'esperienza è senza dubbio inferiore, ma l'abbattimento dei costi garantisce un pubblico molto più ampio. Ne è un esempio lampante "Pokemon Go", un'applicazione in AR sviluppata da Niantic e lanciata il 6 luglio 2016, sia per dispositivi Android che iOS. L'interazione dell'utente con gli elementi virtuali è minima, quasi nulla, ma l'originalità e l'utilizzo innovativo della realtà aumentata hanno contribuito all'enorme diffusione dell'applicazione: 100 milioni di download in meno di un mese e l'applicazione più scaricata in assoluto nella prima settimana da Apple Store [Smi16].

1.5. *Mixed Reality (MR)*

La Mixed Reality, o realtà mista, è anche conosciuta come realtà ibrida; entrambe le denominazioni sottolineano la natura composita di questa tecnologia. La MR unisce infatti i punti di forza delle due tecnologie viste precedentemente (VR e AR) per offrire all'utente un'esperienza di una realtà in cui elementi reale e virtuali si mescolano e interagiscono tra

loro, creando una percezione visivamente realistica dell'insieme. Ci troviamo ad un punto intermedio sulla scala del continuo reale-virtuale: gli spazi che vediamo attraverso il visore sono quelli reali ma gli elementi virtuali sono perfettamente inseriti nella realtà circostante, creando la percezione di un mondo costituito da entrambi i tipi di elementi.

L'idea stessa di realtà mista apre le porte a nuove possibilità di utilizzo: essendo gli oggetti rappresentati realisticamente, l'interazione dell'utente con essi può avvenire in modo molto più semplice e intuitivo, quasi come se si trattasse di oggetti concreti (Figura 4, tratta da <https://www.wired.com/2015/04/microsoft-build-hololens>). Generalmente l'interazione è basata sul riconoscimento dei gesti dell'utente, o anche sull'interpretazione dei comandi vocali; grazie a queste caratteristiche, che mirano ad avvicinare il più possibile l'esperienza della realtà mista a quella della vita reale, gli utenti possono interfacciarsi con facilità all'applicazione e la fase iniziale di training si velocizza.



Figura 4 - Esempio di MR visualizzata tramite MS HoloLens

Una delle possibili applicazioni della MR consiste nel design e nella visualizzazione di modelli 3D: potendo creare, modificare e spostare ologrammi nello spazio si superano notevoli problemi come il costo per la produzione dei modelli fisici o la rappresentazione dei modelli su schermi bidimensionali. Tale funzionalità potrebbe essere applicata al design di qualunque tipo di prodotto, ma anche di interi spazi o edifici. Altri esempi di utilizzi verso

cui sono orientati gli sviluppi attuali sono: applicazioni per training che forniscano istruzioni mediante elementi virtuali, una nuova tipologia di gaming che preveda la realtà circostante come parte integrale dell'ambientazione di gioco o anche il design d'interni, di cui ci occuperemo nello specifico più avanti.

Più in generale, la Mixed Reality sta aprendo le porte ad un nuovo tipo di interfacce utente: quando qualsiasi tipo di informazione può essere rappresentata tramite un ologramma, sia bidimensionale che tridimensionale, gli schermi diventano superflui. Si sta concretizzando l'ipotesi di un futuro in cui uno spazio di lavoro non debba essere fornito di monitor, ma venga creato e personalizzato dall'utente all'interno della realtà mista. In tal caso non sussisterebbe neanche il vincolo degli schermi rettangolari a cui siamo abituati: le informazioni potrebbero essere rappresentate nel modo più consono a seconda dei casi, in qualunque forma e in maniera bidimensionale o tridimensionale. Come conseguenza di tale cambiamento, è stata ipotizzata una totale rivoluzione del web: da documenti ipertestuali bidimensionali, si potrebbe passare a spazi ed elementi virtuali accessibili tramite internet [Alg15]. Queste idee al momento rimangono su un piano ipotetico, ma forse non sono poi così remote se consideriamo la rapida evoluzione delle tecnologie per la realtà mediata e il notevole interesse che esse hanno suscitato negli ultimi anni.

2. VISORI E TECNOLOGIE

2.1. *Visori per la realtà virtuale*

Tra le varie categorie di visori, quelli per la realtà virtuale sono ad oggi i più diffusi e conosciuti. Il loro successo è dovuto in gran parte al loro impiego nel campo dell'intrattenimento, nel quale sono state investite notevoli somme negli ultimi anni. Per la stessa ragione sono numerosi i modelli disponibili sul mercato, con grandi differenze sia riguardo le tecnologie utilizzate che economiche. Di seguito vediamo alcuni esempi di dispositivi, per evidenziare le differenze e le potenzialità delle principali tipologie di visore.

2.1.1. Oculus Rift

Il primo prototipo dell'Oculus Rift è stato finanziato grazie ad una campagna di crowdfunding, iniziata pochi mesi dopo la fondazione dell'azienda Oculus VR, nel 2012. La campagna ebbe enorme successo, raccogliendo 2,5 milioni di dollari: uno dei primi segni dell'attenzione del pubblico nei confronti della Virtual Reality. Dopo la distribuzione limitata dei primi prototipi (Development Kit 1 nel 2013 e Development Kit 2 nel 2014), nel marzo del 2016 è stata rilasciata la versione definitiva dell'Oculus Rift (Figura 5, tratta da <http://www.forbes.com/sites/davidewalt/2016/03/28/oculus-rift-review-the-beginning-of-the-age-of-vr/#11c844861793>), disponibile per tutti gli utenti a un prezzo di circa 600 dollari. La Oculus VR è stata acquistata nel 2014 da Facebook per la somma di 2 miliardi di dollari.



Figura 5 - Oculus Rift

Essendo un visore specifico per la realtà virtuale, l'Oculus Rift occlude completamente la visione dello spazio circostante. Il visore è composto da due schermi con tecnologia OLED con una risoluzione di 1080x1200 per ogni occhio, che garantiscono un campo di visione di 110° in diagonale con una frequenza di aggiornamento di 90 Hz. La posizione dell'utente all'interno dello spazio viene riconosciuta tramite un sistema di tracking chiamato "Constellation": il visore è provvisto di LED a infrarossi predisposti per illuminarsi in uno specifico pattern. Un sensore, posto generalmente sulla scrivania dell'utente, riconosce tale pattern e individua la posizione della testa dell'utilizzatore. Le maggior parte delle applicazioni per Oculus Rift oggi disponibili afferiscono al campo del gaming, motivo che giustifica il notevole successo economico ottenuto, ma sono in fase di sviluppo altre applicazioni nei campi del cinema e del social, come anche di carattere professionale.

2.1.2. HTC Vive

Il visore HTC Vive (Figura 6, tratta da <http://www.wearable.com/headgear/the-best-ar-and-vr-headsets>), sviluppato da HTC insieme a Valve e rilasciato ad aprile 2016, presenta molti punti in comune con l'Oculus Rift, sia dal punto di vista tecnico che da quello estetico. Anche in questo caso infatti lo schermo è composto da due display OLED con risoluzione di 1080x1200, angolo di visione diagonale di 110° e frequenza di aggiornamento di 90 Hz. Anche il sistema di rilevamento della posizione è analogo a quello dell'Oculus, ma nel caso dell'HTC Vive è realizzato tramite due stazioni base che devono essere posizionate in due punti diversi della stanza e che rilevano la posizione dell'utente tramite appositi fotosensori e l'emissione di laser a infrarossi.



Figura 6 - HTC Vive

Una delle maggiori differenze rispetto all'Oculus Rift è la presenza di una telecamera frontale, che fornisce all'utente informazioni sullo spazio circostante. Tale funzionalità è sfruttata però in maniera limitata: è infatti utilizzata esclusivamente nell'applicazione di sistema "Chaperone", un sistema di sicurezza che avverte l'utente in caso di eventuali ostacoli o di oggetti in movimento.

2.1.3. Google Cardboard

Un'alternativa decisamente meno costosa ai visori appena visti è stata presentata da Google nel 2014: si tratta di una semplice scatola di cartone assemblabile dall'utente stesso, in cui sono fissate delle lenti focali da 14 mm, un magnete o una leva conduttiva (a seconda della versione) per il controllo dello schermo e un elastico o del velcro (Figura 7, tratta da <https://www.nrc.nl/nieuws/2016/03/28/allemaal-aan-de-bril-a1476125>). All'interno del visore va poi inserito il proprio smartphone: attraverso una qualunque applicazione compatibile con Google Cardboard l'immagine viene suddivisa in due e distorta, in modo da bilanciare la distorsione causata dalle lenti. In questo modo l'utente ha a disposizione un visore attraverso cui sperimentare la realtà virtuale ad un costo minimo, sfruttando la tecnologia dello smartphone che già possiede.

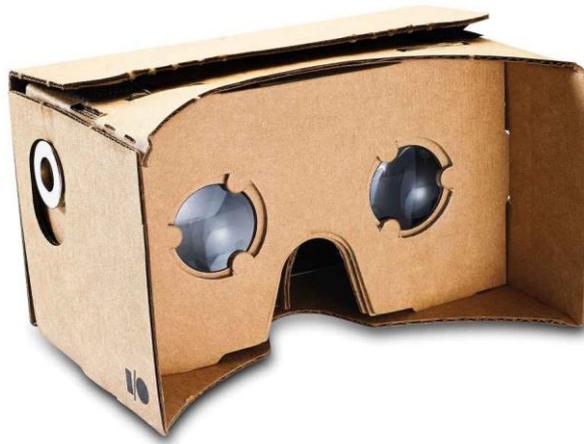


Figura 7 - Google Cardboard

Senza dubbio la caratteristica di rilievo di questo visore è il costo ridotto: il prezzo ufficiale al momento è di 15 dollari. Nei primi 19 mesi dal lancio sono stati acquistati 5 milioni di dispositivi e sono state pubblicate più di 1000 applicazioni compatibili, con 25 milioni di download. Tale successo ha spinto Google non solo a lanciare una nuova versione di Cardboard nel 2015, con modifiche al design per renderlo compatibile agli schermi più ampi degli smartphone delle ultimi generazioni, ma anche e soprattutto ad annunciare una nuova piattaforma migliorata per la VR chiamata “Daydream”, il cui lancio è previsto per la fine del 2016.

2.1.4. Samsung Gear VR

Gear VR è un visore per la realtà virtuale realizzato dalla Samsung in collaborazione con Oculus e rilasciato nel 2015 (Figura 8, tratta da <http://www.samsung.com/us/mobile/virtual-reality/gear-vr/gear-vr-sm-r322nzwaxar>). L'idea alla base del Samsung Gear VR è la stessa di Google Cardboard: la visualizzazione della realtà virtuale è realizzata tramite smartphone, mentre il visore vero e proprio si occupa di interfacciare l'occhio umano allo schermo piatto del telefono. In particolare, Gear VR è compatibile soltanto con determinati smartphone, ovvero quelli appartenenti alle ultime generazioni della linea Galaxy di Samsung (Note 5, S6, S6 Edge, S6 Edge+, S7 e S7Edge).



Figura 8 - Samsung Gear VR

A differenza di Cardboard, Gear VR include al suo interno diversi controlli e sensori che integrano la tecnologia offerta dallo smartphone, per offrire una migliore esperienza all'utente. Ai due lati del dispositivo sono infatti presenti un touchpad per la navigazione dello schermo e dei controlli per la regolazione del volume, l'accesso alla home e alla schermata precedente, ed è possibile regolare la messa a fuoco tramite un rotella posta in cima al visore. Inoltre, è stata integrata nel dispositivo un'unità di misura inerziale (IMU) per il riconoscimento di movimenti rotazionali tramite oscilloscopi e accelerometri che sostituisce quella già presente nello smartphone per ottenere una maggiore precisione.

2.2. Visori per la realtà aumentata

I visori per la realtà aumentata sono decisamente meno numerosi e di minore successo, in parte perché soppiantati dall'utilizzo degli smartphone come supporto all'AR. Ci limitiamo ad esaminare le caratteristiche del più noto tra i vari visori prodotti, Google Glass.

2.2.1. Google Glass

Google Glass (Figura 9, tratta da <http://techjunkie.in/gadgets/google-glasses>), il visore per la realtà aumentata prodotto da Google, è tristemente noto per lo scarso successo di distribuzione. Annunciato nel 2012 e rilasciato per la prima volta nel 2013 per un costo di

1500 dollari, il dispositivo ha attraversato due fasi sperimentali chiamate “Explorer” prima di essere messo fuori commercio nel 2015. Confrontato con gli altri visori, Google Glass risulta particolarmente leggero e poco ingombrante: è costituito da una sottile montatura (disegnata dalla Luxottica) in cui è incorporato un touchpad per il controllo delle funzionalità e da un piccolo display a cristalli liquidi illuminato a led.



Figura 9 - Google Glass

L'utente può interagire con il visore attraverso il touchpad laterale, ma anche con movimenti della testa e degli occhi o tramite comandi vocali. Le principali funzionalità del dispositivo sono la possibilità di scattare foto e registrare video, di ottenere informazioni di navigazione verso un dato luogo, effettuare ricerche in internet, impostare eventi e sveglie o effettuare videochiamate. Una delle ragioni che viene considerata causa del fallimento di Google Glass è il mancato adempimento di uno scopo preciso: il dispositivo offre in generale le stesse funzionalità ottenibili tramite un comune smartphone, senza offrire soluzioni a problemi specifici. Al contrario, l'utente è obbligato ad interfacciarsi con un nuovo tipo di tecnologia a cui non è abituato senza riceverne particolari benefici [Alt15]. Un altro notevole problema è quello della privacy: un utilizzatore di Google Glass potrebbe potenzialmente scattare foto e registrare video in qualsiasi momento, senza il consenso delle persone circostanti. Conseguentemente, l'utilizzo del dispositivo è stato vietato in diversi luoghi pubblici, come cinema o casino.

2.3. *Visori per la realtà mista*

La commercializzazione dei visori per la realtà mista è decisamente più recente rispetto alle due categorie precedentemente analizzate, anche se le ricerche e gli sviluppi riguardanti questo ambito sono in corso già da diversi anni. Al momento i visori per Mixed Reality sono ancora preclusi a gran parte del pubblico, a causa dei costi elevati, ma dato il forte interesse che hanno suscitato ne è prevista una rapida diffusione, con un conseguente abbassamento del prezzo di acquisto. Di seguito verrà analizzato il visore utilizzato per questo lavoro di tesi, Microsoft HoloLens, ed il suo principale competitor, Meta.

2.3.1. Microsoft HoloLens

HoloLens (Figura 10, tratta da <http://pocketnow.com/2015/01/21/windows-holographic>) è il visore per la realtà mista sviluppato e prodotto da Microsoft e rilasciato a marzo 2016. Essendo questo il dispositivo selezionato per lo sviluppo della nostra applicazione, ne analizzeremo le caratteristiche tecniche in dettaglio nel capitolo successivo; in questo paragrafo ci limiteremo a descriverne le funzionalità principali e le differenze con i vari competitor. Il visore è montato su una fascia regolabile che ne distribuisce il peso sulla testa dell'utilizzatore ed è leggermente oscurato. Al suo interno sono presenti due lenti combinatorie sulle quali vengono proiettate le immagini. Ai due lati del visore sono presenti vari comandi per la regolazione del volume e della luminosità, come anche due altoparlanti 3D, posizionati in prossimità delle orecchie dell'utente, che simulano la provenienza dei suoni da punti precisi dello spazio.



Figura 10 - Microsoft HoloLens

Molte delle funzionalità caratteristiche dell'HoloLens si basano sul riconoscimento dello spazio reale, in modo che gli elementi virtuali possano interagire con esso. Sono presenti perciò numerosi sensori, tra cui: quattro sensori per il riconoscimento dell'ambiente, uno per la profondità, uno per la luminosità, una fotocamera da 2 MP, quattro microfoni e un'unità di misura inerziale (IMU) costituita da accelerometro, giroscopio e magnetometro. Inoltre è presente una HPU (Holographic Processing Unit), ovvero un coprocessore progettato appositamente da Microsoft per l'elaborazione dei dati di input provenienti dai vari sensori e per la realizzazione del riconoscimento spaziale, gestuale e vocale. L'utente può interagire con il visore in tre modi differenti: attraverso la direzione dello sguardo, i gesti della mano e i comandi vocali. Si fa riferimento all'insieme di queste tre modalità di input come GGV (gaze, gesture, voice).

2.3.2. Meta Development Kit 2

Meta Development Kit 2 (Figura 11, tratta da <https://blog-vrheadset.blogspot.it/2016/07/meta-pokemon-go-akan-jauh-lebih-baik.html>) è uno dei principali competitor dell'HoloLens: sia il design che le funzionalità sono piuttosto simili ed entrambi i dispositivi offrono all'utente un'esperienza di realtà mista, inserendo ologrammi all'interno dello spazio reale. L'omonima azienda Meta ha dato il via allo sviluppo del primo dispositivo nel 2013 grazie a una campagna di crowdfunding che ha raccolto circa 200.000 dollari; il primo dispositivo, Meta Development Kit 1, è stato rilasciato nel 2015. Il secondo, attualmente

l'ultimo modello, è stato lanciato a febbraio del 2016 ancora in edizione per sviluppatori ad un prezzo di circa 1000 dollari.



Figura 11 - Meta Development Kit 2

Una differenza fondamentale tra HoloLens e Meta è che se il primo è un dispositivo completamente autonomo e indipendente, il secondo deve necessariamente essere collegato ad un computer, costringendo l'utente ad un campo di movimento particolarmente ridotto e compromettendo l'esperienza della realtà mista. D'altro canto, il visore del Meta presenta un display con un campo di azione molto più ampio (90° in diagonale), che consente un'immersione più completa nella realtà mista, mentre nel caso dell'HoloLens gli ologrammi sono visibili solo in un'area ridotta posta al centro del campo visivo. Inoltre, il riconoscimento dei gesti in Meta è particolarmente avanzato e permette all'utente di interagire con gli ologrammi in maniera immediata ed intuitiva: si possono afferrare o spostare gli oggetti con la gestualità tipica della vita reale, o modificare lo zoom con il gesto di "pinch" ormai tipico di tutti i touch screen [Pra15].

3. MICROSOFT HOLOLENS: SPECIFICHE TECNICHE, STRUMENTI DI SVILUPPO E FUNZIONALITÀ

3.1. Caratteristiche tecniche e di sviluppo

In questo capitolo verrà analizzato in dettaglio il visore MS HoloLens, esaminandone l'hardware ed evidenziandone aspetti innovativi e potenzialità. Verranno quindi indicati quali sono i requisiti di sistema richiesti per lo sviluppo e quali gli strumenti da utilizzare.

3.1.1. Caratteristiche hardware

Il modello di Microsoft HoloLens a cui si fa riferimento è l'HoloLens Development Edition, ovvero il primo mai messo in commercio. Rilasciato a marzo 2016, inizialmente era destinato ad un target specifico di programmatori e acquistabile a un costo di 3000 dollari solo tramite apposita candidatura, ma a partire dal 2 agosto le vendite sono state aperte al pubblico.

HoloLens è un dispositivo completamente autonomo: questo vuol dire che tutte le unità per la computazione, i vari sensori ed i display per la visualizzazione degli ologrammi sono racchiusi all'interno del visore. Sono presenti un SoC (system on a chip) Intel Cherry Trail x86, una IMU e una HPU; gran parte del carico di lavoro è svolta dal SoC, con il supporto della HPU per quanto riguarda l'interpretazione dei dati provenienti dai vari sensori. La HPU (Holographic Processing Unit, attualmente 1.0), sviluppata appositamente per l'elaborazione dei dati riguardanti il riconoscimento spaziale e gestuale, può eseguire circa un trilione di calcoli al secondo, anche se al momento lavora soltanto al 50% delle sue potenzialità.

Le informazioni inviate alla HPU sono raccolte da vari sensori presenti sul dispositivo: l'IMU, quattro telecamere di riconoscimento ambientale, un sensore di profondità, uno di luminosità, una serie di quattro microfoni ed una telecamera da 2 megapixel. In particolare, la IMU (inertial measurement unit) si occupa di raccogliere gli input per il riconoscimento

dei movimenti della testa dell'utente attraverso l'utilizzo combinato di giroscopi, accelerometri e magnetometri [Dea16]. Come si può notare dalla Figura 12 (tratta da <https://www.computerbase.de/bildstrecke/74181/1>) i vari segnali provenienti dai sensori vengono raccolti ed elaborati dalla HPU, prima di essere trasmessi al SoC.

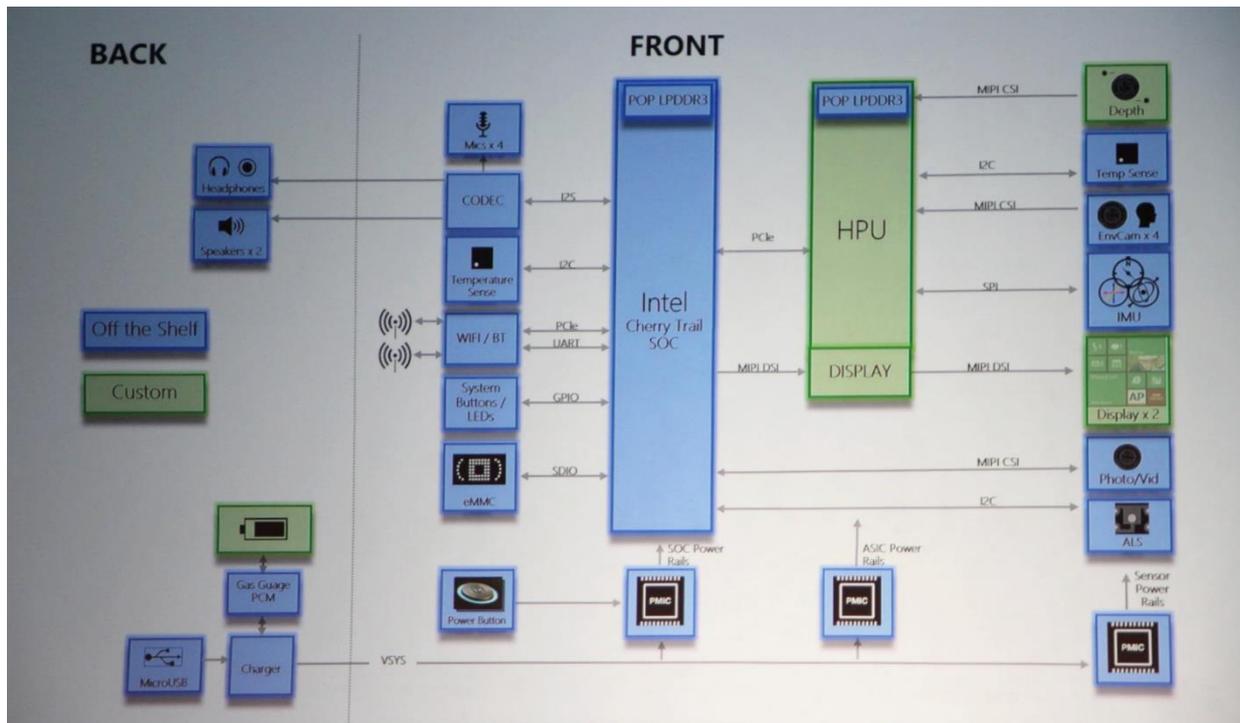


Figura 12 - Architettura del sistema di MS HoloLens.

3.1.2. Strumenti di sviluppo e requisiti

Per lo sviluppo di applicazioni per l'HoloLens è richiesto l'utilizzo di Visual Studio 2015 con Update 3, che comprende i Tools in versione 1.4 e il Windows 10 SDK; è consigliato lavorare con Windows 10, ma quasi tutte le funzionalità sono supportate anche in Windows 8.1, 8 e 7. È disponibile un emulatore (HoloLens Emulator) per eseguire le applicazioni sulla piattaforma Windows Holographic in una macchina virtuale, in mancanza del dispositivo fisico. Per utilizzarlo sono richiesti una CPU a 64 bit, un minimo di 8 GB di RAM e una versione di Windows 10 che supporti la funzionalità Hyper-V per la virtualizzazione, ovvero Pro, Enterprise o Education.

Per lo sviluppo delle scene tridimensionali e per la manipolazione degli ologrammi viene utilizzato Unity, un software multiplatforma utilizzato principalmente per lo sviluppo di videogiochi. In particolare, per la realizzazione di applicazioni in realtà mista per HoloLens è stata rilasciata la versione gratuita Unity HoloLens Technical Preview. Grazie a Unity è possibile partire dalla costruzione della scena 3D per poi trasferire il progetto implementato in C# in Visual Studio.

3.1.3. Piattaforma di esecuzione e sistema operativo

Microsoft HoloLens è il primo dispositivo ad utilizzare la piattaforma Windows Holographic, sotto il sistema operativo Windows 10. Si tratta di una piattaforma sviluppata da Microsoft appositamente per la realtà mista, e quindi per l'esecuzione di applicazioni che integrino elementi virtuali (i cosiddetti ologrammi) all'interno dello spazio reale.



Figura 13 - Esempio di shell visualizzata attraverso MS HoloLens. Immagine tratta da:

La shell dell'HoloLens (Figura 13, tratta da <http://uk.businessinsider.com/microsoft-hololens-augmented-reality-headset-roundup-2015-10>) è composta dall'insieme del mondo reale e di specifici ologrammi: il menu start è rappresentato dall'ologramma di un rettangolo bidimensionale che è possibile muovere nello spazio, attraverso il quale lanciare

le varie applicazioni. Una volta selezionata un'applicazione, comparirà nello spazio circostante il corrispondente "asset", che può essere rappresentato sia come una finestra 2D che come un oggetto tridimensionale; facendo "tap" sull'asset, si avvia l'esecuzione dell'applicazione scelta. Grazie alle API della piattaforma Windows Holographic, adesso incluse nella Universal Windows Platform e supportate come standard, è possibile implementare facilmente le funzionalità per la realtà mista in qualsiasi Universal Windows App.

3.2. *Funzionalità ed interazione*

L'idea alla base della realtà mista è la rappresentazione realistica di elementi virtuali: questo vuol dire che gli ologrammi devono essere integrati nello spazio circostante e che l'utente deve poterli gestire in maniera simili a come farebbe con oggetti reali. Per realizzare tali funzionalità è opportuno comprendere come è strutturata in Windows Holographic la gestione dei segnali di input e come avviene il riconoscimento spaziale e gestuale.

3.2.1. Tipologie di input: sguardo, gesti e comandi vocali

Per rendere l'esperienza di realtà mista piacevole ed intuitiva per l'utente, i comandi di input sono strutturati in modo da essere molto simili alle interazioni che siamo abituati ad avere con gli oggetti reali. I comandi possono essere di tre tipi, riassunti in inglese in GGv (gaze, gesture, voice). Il comando "gaze" è il più basilare dei tre: attraverso il riconoscimento della posizione della testa dell'utente viene individuato il punto centrale del suo campo visivo, che viene utilizzato come puntatore. In molte applicazioni infatti in tale punto è visibile un pallino luminoso oppure un piccolo cerchio che evidenzia la posizione corrente del cursore.

La posizione del cursore viene quindi utilizzata dagli altri comandi per svolgere diverse funzionalità; ad esempio, una volta raggiunta la posizione desiderata tramite la direzione dello sguardo, è sufficiente abbassare un dito (comando "air tap") per simulare il funzionamento del click. Il gesto "air tap" è in realtà diviso in due fasi: la prima in cui l'indice della mano viene abbassato e la seconda, in cui si ritorna alla posizione iniziale. Fermando

il dito a metà del gesto è possibile ottenere altri comportamenti, come il trascinamento o l'equivalente del tasto destro di un mouse. Il secondo ed ultimo gesto riconosciuto dall'HoloLens è quello di "bloom", effettuato aprendo tutte le dita della mano con il palmo rivolto verso l'alto. Questo gesto assume significati differenti a seconda dello stato dell'esecuzione: se ci si trova nella schermata di home serve a richiamare il menu start, altrimenti viene usato per spostare l'applicazione correntemente in esecuzione in background e tornare alla home.

In ultimo, è possibile interagire con il visore tramite comandi vocali. Ovviamente non è possibile accedere a tutte le funzionalità utilizzando esclusivamente la voce, ma in molte applicazioni i comandi vocali costituiscono un'utile integrazione, soprattutto quando esprimere il comando desiderato tramite gesti fisici sarebbe particolarmente complicato o noioso. Ad esempio, vengono spesso utilizzati come alternativa alla digitazione di testo, che altrimenti obbligherebbe l'utente a fare "tap" tasto per tasto su una tastiera virtuale. Attualmente il riconoscimento vocale è disponibile soltanto in lingua inglese.

3.2.2. Spatial Mapping

Per consentire l'integrazione tra elementi virtuali e spazio reale deve essere possibile conoscere ed elaborare gli spazi circostanti all'utente. Grazie all'utilizzo dei sensori di luminosità, di profondità e d'ambiente, Windows Holographic mette a disposizione degli sviluppatori delle API apposite; questo processo viene definito "Spatial Mapping". Attraverso specifiche funzioni, è possibile esplicitare con quale definizione dovrà essere effettuata la ricostruzione dello spazio fisico, specificando quanti triangoli bidimensionali impiegare per rappresentare le superfici comprese in un metro cubo di spazio.

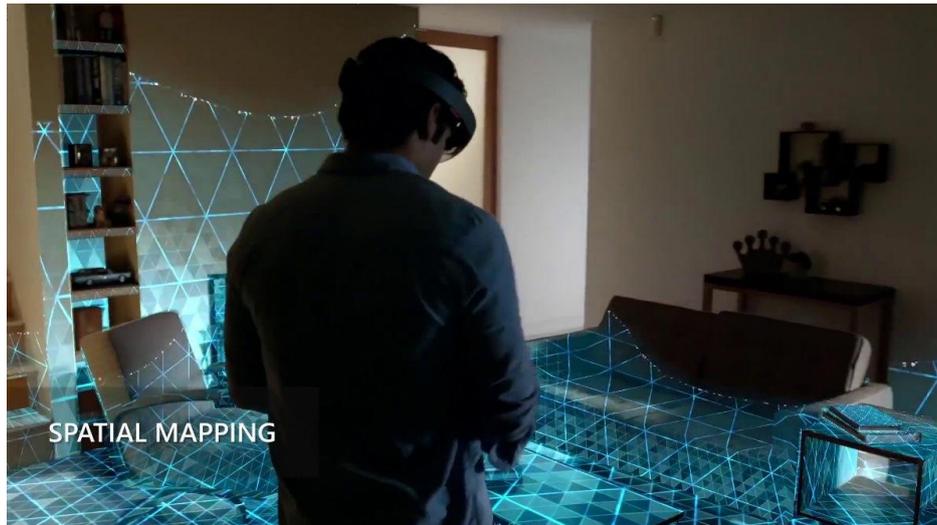


Figura 14 - Esempio di Spatial Mapping effettuato tramite MS HoloLens

L'HoloLens offre la possibilità di memorizzare gli spazi esaminati in modo da evitare di dover esplorare lo stesso spazio ad ogni avvio, anche se la mappatura viene continuamente aggiornata durante l'esecuzione per registrare eventuali cambiamenti dell'ambiente; la mappa dello spazio salvata in memoria viene identificata tramite la rete Wi-Fi a cui il dispositivo è connesso. Una volta ottenuta la rappresentazione tridimensionale dell'ambiente, essa può essere utilizzata per posizionare gli ologrammi al suo interno: ad esempio, per fare in modo che un particolare ologramma possa essere posto solo dove sia presente una superficie piatta abbastanza ampia da ospitarlo (Figura 14, tratta da <http://arnews.tv/microsoft-hololens-spatial-mapping>). Inoltre, grazie allo Spatial Mapping, gli ologrammi possono essere rappresentati con maggiore realismo, nascondendone alcune parti quando esse dovrebbero essere nascoste alla vista da oggetti reali o anche da altri ologrammi presenti nello spazio.

3.2.3. Spatial Sound

Un'altra utile funzionalità offerta da Windows Holographic è la possibilità di realizzare lo "Spatial Sound", ovvero degli effetti sonori legati alla posizione degli ologrammi che mirano a rendere la percezione acustica il più verosimile possibile. Grazie all'analisi combinata della posizione della fonte sonora e della testa dell'utente è possibile riprodurre i suoni in maniera tridimensionale attraverso gli appositi altoparlanti 3D, posti ai due lati del visore.

In questo modo, la percezione degli oggetti diventa più completa e realistica, poiché la fonte sonora viene ricollegata immediatamente all'ologramma corrispondente, in maniera completamente istintiva. Un possibile utilizzo di questa funzionalità potrebbe essere far percepire all'utente la presenza di ologrammi che al momento di trovano al di fuori del suo campo visivo, o anche utilizzare i suoni come sistema di notifica che richiami l'attenzione dell'utente su un preciso elemento.

3.2.4. Condivisione degli ologrammi

Nel caso si abbia l'opportunità di lavorare con più di un visore simultaneamente, HoloLens offre la possibilità di condividere l'esecuzione di un'applicazione su più di un dispositivo, in modo da poter avere accesso allo stesso ambiente virtuale. Tale funzionalità è realizzata tramite la condivisione di "ancore spaziali": un'ancora consiste in delle coordinate spaziali che identificano univocamente un punto all'interno dell'ambiente reale. Una volta che il primo utente ha avviato l'esecuzione dell'applicazione e l'ologramma è stato posizionato in un punto preciso dello spazio, viene individuata la corrispondente ancora spaziale che viene poi condivisa con il secondo dispositivo. In tal modo, il secondo HoloLens può utilizzare le coordinate rappresentate nell'ancora per posizionare lo stesso ologramma nello spazio e quindi riprodurre lo stesso ambiente virtuale del primo utente.



Figura 15 - Esempio di condivisione di ologrammi tra più visori HoloLens

La condivisione degli ologrammi (Figura 15, tratta da <https://www.youtube.com/watch?v=Ey8yBgWiqtg>) risulta utile nel caso in cui uno dei due utilizzatori voglia effettuare una dimostrazione, e quindi gli sia necessario visualizzare l'ambiente virtuale ma al contempo darne accesso ad uno o più ulteriori utenti. Ancora più interessante, è la possibilità di un'interazione simultanea di due o più utenti con lo stesso ologramma, ovvero di un lavoro collaborativo in cui tutti gli utenti abbiano accesso completo alle varie funzionalità, proprio come succederebbe nel caso di un ambiente di lavoro reale.

Grazie alla condivisione degli ologrammi si può pensare a soluzioni particolarmente innovative: ad esempio, scenari di gaming multi-giocatore, in cui la scena di gioco venga condivisa tra più utenti, oppure applicazioni che sfruttino la condivisione per realizzare una "socializzazione" in cui gli utenti non siano isolati ma possano scambiarsi informazioni e messaggi durante l'utilizzo.

3.3. Stato attuale dello sviluppo per MS HoloLens

Trattandosi di una tecnologia completamente nuova ed ancora in fase di sviluppo, sono pochi gli esempi di applicazioni già realizzate a cui è possibile fare riferimento. Questa mancanza è compensata da parte di Microsoft con la condivisione di diversi progetti il cui codice è messo a disposizione degli sviluppatori. La community di sviluppatori per Windows Holographic al momento non è molto estesa ma, anche grazie alla recente apertura al pubblico della vendita dell'HoloLens, è in continua crescita e particolarmente attiva.

Microsoft mette a disposizione degli sviluppatori una "Academy", ovvero un insieme di tutorial che guidano l'utente nei primi passi della realizzazione di un'applicazione per HoloLens, ed un forum grazie al quale gli utenti possono confrontarsi e chiedere consiglio o condividere il codice dei propri progetti. Uno dei primi progetti, sviluppato dal team di Microsoft stesso subito dopo il lancio dell'HoloLens, è "Galaxy-Explorer", un'applicazione che consente di visualizzare una riproduzione 3D della Via Lattea e fornisce all'utente informazioni sui vari pianeti e costellazioni (Figura 16, tratta da

<http://www.winbeta.org/news/microsoft-releases-hololens-galaxy-explorer-app-source-code>). Galaxy-Explorer è stato sviluppato in completa collaborazione con il pubblico: selezionato tra varie idee proposte dagli utenti, è stato realizzato in circa due mesi e una volta completato il codice è stato reso pubblico, consentendo agli sviluppatori di proporre eventuali estensioni, ma anche di prendere spunto dal codice del progetto per lavori propri.



Figura 16 - Il progetto Galaxy-Explorer, pubblicato dal team Microsoft

Tutti gli utenti sono invitati da Microsoft a condividere i propri progetti con gli altri sviluppatori attraverso il forum, in modo da incoraggiare lo scambio di idee e facilitare il lavoro agli utenti che si avvicinano a Windows Holographic per la prima volta. Oltre al codice del Galaxy-Explorer, Microsoft mette a disposizione degli utenti HoloToolkit, ovvero una collezione di script e di componenti specifici per Windows Holographic, utili per velocizzare e semplificare il processo di sviluppo delle applicazioni. In particolare, HoloToolkit include strumenti per la gestione degli input, dello Spatial Mapping, dello Spatial Sound e della condivisione degli ologrammi.

4. APPLICAZIONE: PREMESSE E OBIETTIVI

4.1. Introduzione e premesse allo sviluppo dell'applicazione

Analizziamo adesso quali possono essere le potenzialità della Mixed Reality in un caso concreto di utilizzo, nello specifico in relazione all'interior design. Con la collaborazione di Poltronesofà è stata realizzata un'applicazione che aiuti gli utenti nella scelta di un elemento di arredamento, liberandoli grazie alla virtualizzazione del vincolo dell'acquisto e del trasporto fisico del prodotto.

4.1.1. Campo di interesse dell'applicazione

La nostra applicazione rientra nel campo del design d'interni: la Mixed Reality è particolarmente adatta a questo tipo di utilizzo poiché consente di avere percezione del mondo reale e allo stesso tempo degli elementi virtuali generati dal visore. In questo modo è possibile realizzare applicazioni che mostrino agli utenti il risultato finale di un progetto di arredamento, unendo lo spazio attuale a disposizione del cliente alla riproduzione virtuale degli elementi di arredo che verranno inseriti in seguito.

Grazie alla tecnologia di Microsoft HoloLens è possibile andare oltre il semplice inserimento degli elementi, realizzando una vera e propria interazione tra l'utente e gli ologrammi: la situazione che si vuole simulare è quella di un cliente che, scelto un particolare mobile, lo sposta e lo ruota all'interno della stanza per trovarne la posizione ottimale. Con la realtà mista, però, il momento della "prova" può precedere quello dell'acquisto: senza muoversi da casa, l'utente ha a disposizione il catalogo completo e può provare a posizionare gli elementi nel proprio ambiente, confrontando magari i vari colori e materiali per scegliere quello che più lo convince. Inoltre l'utente non è più obbligato ad acquistare e trasportare fisicamente gli elementi di arredo senza essere sicuro dell'effetto finale, ma può liberamente provare a visualizzare tutti i modelli in tutte le possibili combinazioni prima di concludere effettivamente l'acquisto.

4.1.2. Requisiti del cliente

Il cliente con cui abbiamo collaborato per la realizzazione di questa applicazione è Poltronesofà S.p.a., azienda nata nel 1995 a Forlì e oggi uno dei marchi leader in Italia nella vendita di elementi di arredamento, con più di 190 punti vendita. È stata richiesta un'applicazione che consentisse ai clienti di scegliere il divano desiderato tra varie opzioni e di posizionarlo nell'ambiente a disposizione. I modelli tridimensionali utilizzati per lo sviluppo dell'applicazione sono stati forniti da Poltronesofà e corrispondono a modelli di divani attualmente in vendita e consultabili da catalogo (Figura 17, tratta da <http://www.poltronesofa.com/en-us/Sofas>). Un'altra possibilità di utilizzo previsto per l'applicazione è di fornirla in dotazione a chi si occupa di pubblicizzare gli arredamenti presso i clienti o ai designer di interni professionisti, in modo che i clienti che vogliono acquistare un divano possano usufruirne senza essere obbligati all'acquisto del visore.

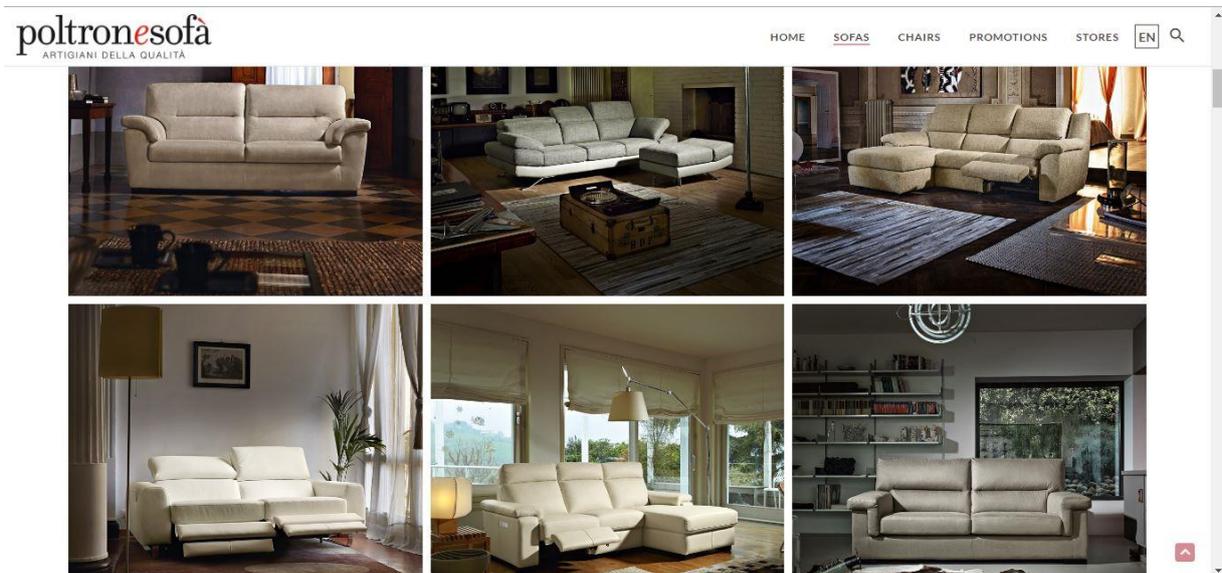


Figura 17 - Campione di modelli dal catalogo di Poltronesofà

In tal modo un unico visore, su cui sia stata installata l'applicazione, potrebbe essere usato svariate volte presso vari clienti, grazie alla capacità dell'applicazione di adattarsi a qualunque spazio. Al momento questo è il tipo di impiego che si ritiene più probabile e facilmente attuabile, essendo l'HoloLens una tecnologia ancora poco diffusa e con un prezzo non particolarmente accessibile. Utilizzare lo stesso visore in più situazioni differenti,

quindi, consente di ammortizzare il prezzo e permette all'azienda di offrire ai suoi clienti un servizio innovativo ed interessante, che può rappresentare un importante incentivo all'acquisto con un costo aggiuntivo praticamente nullo.

Non è escluso che in futuro, con la diffusione delle tecnologie per la realtà mista, sia possibile destinare l'applicazione ad un target di utenti privati, che la utilizzino autonomamente sul loro visore personale. In una tale eventualità, l'applicazione potrebbe essere estesa per garantire una migliore personalizzazione dell'esperienza, adattando l'offerta dei divani alle possibilità dell'utente particolare e magari aiutandolo nella scelta con suggerimenti calibrati sulle sue preferenze.

4.1.3. Obiettivi dell'applicazione

Analizziamo qui, sotto forma di lista, i punti fondamentali che dovranno essere raggiunti durante lo sviluppo dell'applicazione; più avanti ne forniremo una descrizione più dettagliata.

- L'applicazione dovrà presentare all'utente diversi campioni di modelli tra cui l'utente potrà scegliere liberamente, un solo modello per volta.
- L'utente potrà visualizzare il modello richiesto nello spazio sotto forma di ologramma.
- L'utente potrà interagire col l'ologramma attraverso quattro possibili azioni: il posizionamento del divano, la rotazione, la modifica del colore del tessuto e la rimozione del divano dalla scena corrente.
- Le quattro operazioni potranno essere eseguite in qualsiasi ordine e ripetute un numero qualsiasi di volte.
- In seguito all'operazione di rimozione, all'utente verrà ripresentato il campionario dei modelli, per permettergli di fare una nuova selezione e di iniziare una nuova sessione di interazione.
- A discrezione dell'utente, tutte le operazioni finora menzionate potranno essere eseguite tramite comandi vocali, in alternativa ai comandi gestuali.

L'utente potrà quindi effettuare la scelta del modello desiderato da un catalogo delle varie opzioni disponibili e poi interagire con esso nel modo più intuitivo possibile. Dato che i gesti di interazione permessi da HoloLens sono soltanto due ("air tap" e "bloom"), gran parte dei comandi saranno realizzati come bottoni tridimensionali, ovvero anch'essi ologrammi presenti nello spazio insieme al modello del divano. In questo modo sarà possibile invocare tutti i comandi necessari attraverso lo stesso gesto di "tap".

I vari comandi saranno organizzati in vari menu con diversa semantica e non saranno presentati mai contemporaneamente all'utente: in ogni momento dovranno essere visibili solo i bottoni relativi a comandi validi ed applicabili allo stato di esecuzione attuale. I vari set di comandi saranno a loro volta contenuti in un menu, attraverso il quale l'utente potrà decidere in ogni momento di invocarli: ogni set corrisponderà ad una particolare funzionalità, ovvero la selezione del modello, lo spostamento, la rotazione e la scelta del materiale o del colore.

Alternativamente alla selezione del comando tramite "tap", dovrà essere possibile utilizzare qualsiasi funzione tramite comando vocale: il riconoscimento vocale sarà quindi attivo in ogni momento e pronto ad interpretare i comandi dell'utente, attivando la corrispondente funzionalità solo quando la situazione attuale lo consente. Per aiutare l'utente ad associare il comando desiderato alla parola chiave corrispondente, ai bottoni dei vari comandi sarà sovrainpresso il testo che l'utente dovrà pronunciare per ottenere la funzionalità desiderata. Grazie ai comandi vocali dovrà quindi essere possibile utilizzare l'applicazione in tutta la sua potenzialità, dall'inizio alla fine dell'esecuzione, anche senza mai utilizzare i comandi gestuali.

4.2. Strumenti di sviluppo

A causa dell'importante componente grafica delle applicazioni per HoloLens è necessario utilizzare specifici software e ambienti di sviluppo per creare la struttura dell'ambiente virtuale. In particolare, sono stati utilizzati Unity, nella versione per HoloLens, per costruire la scena dell'applicazione e Blender, per la manipolazione dei modelli tridimensionali.

4.2.1. Unity, piattaforma di sviluppo tridimensionale

Unity è un software di sviluppo multi-piattaforma utilizzato principalmente nel campo del gaming. Rilasciato inizialmente nel 2005 ha avuto un notevole successo: nel 2012 era già utilizzato da 1,3 milioni di sviluppatori [Tak12]. Caratteristiche che rendono Unity un ambiente di sviluppo così popolare sono la disponibilità di una versione gratuita, la compatibilità con gran parte degli altri software per il 3D in commercio, ma soprattutto le grandi potenzialità di utilizzo, che consentono agli sviluppatori di realizzare qualunque tipo di videogioco o di applicazione che preveda una scena tridimensionale [Pol13]. In collaborazione con Microsoft, è stata rilasciata una versione specifica per lo sviluppo di applicazioni in Mixed Reality, ovvero la Unity HoloLens Technical Preview, un'estensione di Unity 5.4 Beta. Questa particolare versione di Unity mette a disposizione degli sviluppatori numerose API specifiche per le applicazioni olografiche, e quindi per la gestione degli input gestuali, dei comandi vocali, dello Spatial Mapping o dello Spatial Sound.

4.2.2. Costruzione della scena di un'applicazione

Unity velocizza notevolmente il processo di sviluppo delle applicazioni per HoloLens, permettendo all'utente di costruire graficamente la "scena", ovvero quella che diventerà l'ambiente virtuale una volta messa in esecuzione l'applicazione (Figura 18). Gli elementi fondamentali per la costruzione di una scena sono definiti "GameObject": non hanno caratteristiche specifiche, ma agiscono da contenitori per i vari componenti. Tutti gli elementi che sono presenti all'interno della scena sono infatti organizzati in delle gerarchie, ed ognuno di essi fa riferimento ad uno specifico GameObject, ovvero la radice di quel particolare insieme di elementi. La scena ha una configurazione a foresta: sono presenti diverse radici, tutte distinte e sullo stesso livello gerarchico. Nel caso della nostra applicazione sono sei ed ognuna di esse contiene gli elementi relativi ad un diverso campo semantico: la Camera, le luci, gli elementi grafici, lo Spatial Mapping, il cursore ed i Manager per la gestione degli input.

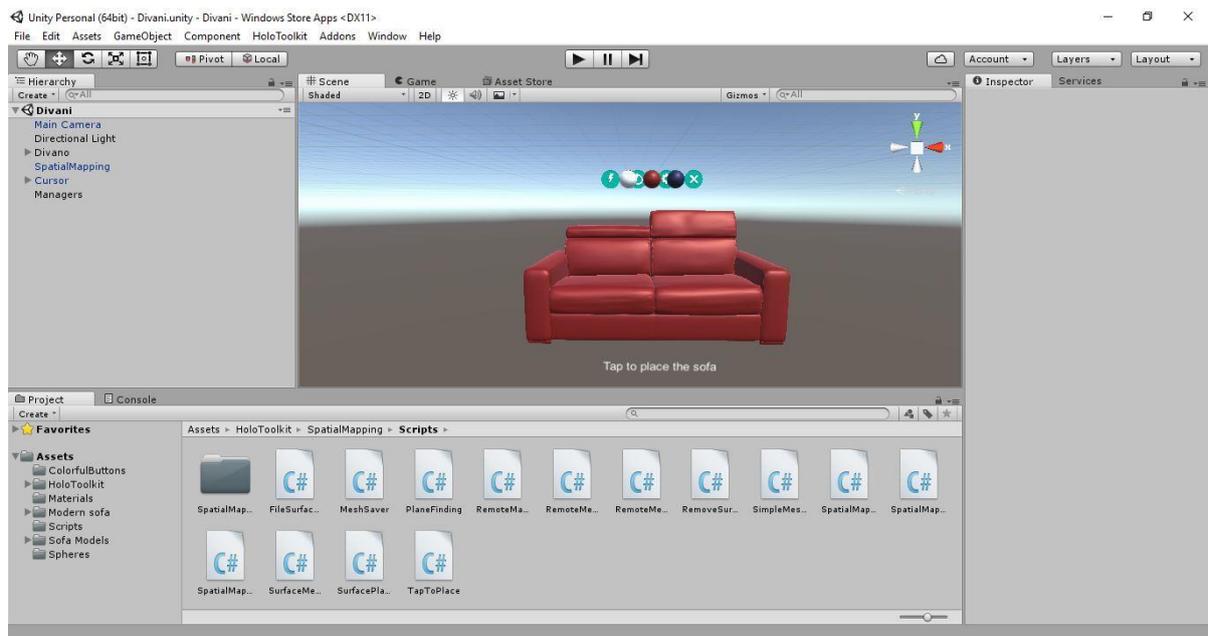


Figura 18 - Esempio di visualizzazione di una scena in Unity

I GameObject possono avere quindi qualsiasi semantica, a seconda dei componenti che contengono: possono rappresentare un oggetto fisico, una fonte di luce, un punto di vista o una fonte sonora. Ad ogni GameObject è sempre associato un componente di tipo Transform, che rappresenta la posizione e l'orientamento nello spazio di tutta la gerarchia di elementi racchiusi in quel GameObject; ogni particolare componente sarà associato poi ad un proprio oggetto Transform, che ne potrà alterare la posizione e l'orientamento rispetto al GameObject contenitore.

Un GameObject fondamentale, che dev'essere necessariamente presente nel progetto di un'applicazione per HoloLens, è la "MainCamera". Gli oggetti di tipo "Camera", in generale, rappresentano il punto di vista dell'utente all'interno dell'ambiente virtuale, ovvero il punto preciso nello spazio in cui si trova l'utente e la direzione con cui visualizza gli ologrammi. È quindi indispensabile che nella scena ci sia almeno un oggetto di questo tipo, per simulare la posizione del visore rispetto agli ologrammi al momento dell'esecuzione. Per ottimizzare la MainCamera per l'HoloLens vanno modificate alcune impostazioni: il background dev'essere settato sul nero (RGBA 0, 0, 0, 0), che il visore interpreterà come trasparente, in modo che intorno agli ologrammi sia visibile lo spazio reale, ed è consigliato posizionare la MainCamera nell'origine (X:0, Y:0, Z:0) per gestire con maggiore facilità la posizione degli

elementi intorno all'utente. La posizione della MainCamera sarà poi aggiornata automaticamente durante l'esecuzione a seconda degli spostamenti dell'utente che indossa il visore.

4.2.2.1. Integrazione del codice nella scena dell'applicazione

Una volta che i GameObject ed i rispettivi componenti sono stati aggiunti alla scena, essi rappresenteranno elementi grafici completamente statici: la dinamicità dell'applicazione, che consente l'interazione dell'utente con gli ologrammi, va realizzata tramite codice. In generale Unity consente di inserire codice scritto sia in C# che in Javascript, ma la compatibilità con Windows Holographic limita la scelta del linguaggio al C#. Ogni file di script che viene creato deve essere associato ad un elemento nella gerarchia della scena, sia esso un GameObject o un singolo componente.

Il tempo di vita dell'esecuzione di uno script è in stretta correlazione con l'elemento a cui è stato associato: se durante l'esecuzione dell'applicazione un elemento viene disattivato perché superfluo, anche i processi relativi agli script ad esso associato verranno terminati. I file contenenti il codice vengono editati esternamente a Unity, in Visual Studio. Tutte le classi realizzate devono essere estensioni della classe base "MonoBehaviour"; due funzioni che generalmente è utile ridefinire quando viene creata una nuova sottoclasse sono "Start()" e "Update()", che saranno invocate rispettivamente all'avvio della scena di gioco e ad ogni aggiornamento del display. In particolare, la funzione "Start()" è separata dal costruttore del componente perché l'inizializzazione del componente stesso generalmente avviene in un momento precedente l'avvio della scena. A queste due funzioni base verranno poi aggiunte tutte quelle utili alla gestione dei vari eventi, per realizzare l'interazione dell'utente con gli elementi della scena.

4.2.3. Blender, software per la manipolazione di modelli 3D

Nello sviluppo di applicazioni olografiche esiste una forte componente di carattere grafico: spesso infatti gli sviluppatori sono affiancati da un team di designer che si occupano di realizzare i modelli 3D sui quali verrà poi costruita l'applicazione. In questo tipo di

applicazioni, l'impatto visivo e l'aspetto estetico degli ologrammi rivestono una particolare importanza nel rendere piacevole l'esperienza dell'utente, per cui è ragionevole delegare tali responsabilità a professionisti della grafica tridimensionale. Nel nostro caso, non è risultato necessario l'intervento di grafici professionisti poiché i modelli tridimensionali sono stati forniti dal cliente, e si è quindi usufruito del lavoro già svolto in precedenza per la realizzazione di vari modelli di divani selezionati dal catalogo Poltronesofà.

Ciononostante, si è dovuto intervenire in minima parte sui suddetti modelli per poterli adattare ai requisiti di Unity; in particolare, un singolo modello non può essere costituito da più di 65000 triangoli o da più di 65000 vertici, caratteristica che inizialmente i modelli forniti non rispettavano. Per ottenere tale risultato è stato utilizzato Blender, un software open-source specifico per la creazione e la modifica di modelli tridimensionali (Figura 19). È stato quindi utilizzato lo strumento "Decimate Modifier" per ridurre la complessità dei modelli dei divani: si è riuscito ad ottenere un numero di vertici e di poligoni sufficientemente basso senza danneggiare la resa visiva degli ologrammi. Inoltre, sempre utilizzando Blender, è stato effettuato il merge dei vari componenti costitutivi del modello in un unico elemento, con lo scopo di semplificarne la gestione una volta importato in Unity.

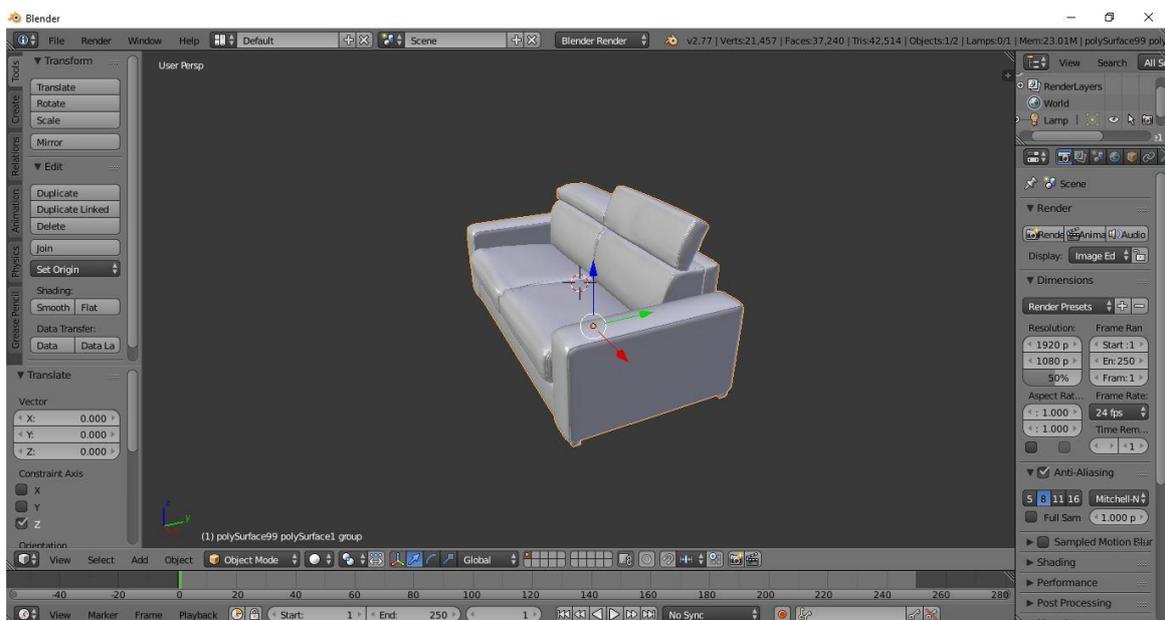


Figura 19 - Esempio di modifica di un modello 3D in Blender

4.2.4. HoloToolkit, collezione di componenti per Windows Holographic

HoloToolkit consiste in una serie di script e componenti precostruiti, realizzati da Microsoft e messi a disposizione degli sviluppatori attraverso un repository GitHub pubblico. L'intento di questo progetto è di semplificare e velocizzare la realizzazione di applicazioni per Windows Holographic, anche in considerazione dello stato preliminare degli sviluppi; per il progetto HoloToolkit è stato infatti adottato il "Microsoft Open Source Code of Conduct", che prevede la totale condivisione del codice e la collaborazione dell'intera community allo sviluppo.

Gli elementi dell'HoloToolkit sono stati ottimizzati per l'utilizzo in Unity: è infatti possibile scaricare dei componenti in formato ".prefab", che dovranno poi essere importati direttamente nella scena di lavoro del progetto Unity e che rappresentano dei GameObject già completi sia di componenti grafici che di script. Gli elementi che sono stati impiegati nella nostra applicazione sono quelli riguardanti l'elaborazione degli input vocali (Speech Manager), il riconoscimento spaziale (Spatial Mapping), il tracciamento dello sguardo dell'utente (Gaze Manager e Cursor) e l'interpretazione dei comandi gestuali (Gesture Manager).

5. ARCHITETTURA E IMPLEMENTAZIONE DELL'APPLICAZIONE

5.1. *Struttura dell'applicazione*

Per aiutare l'utente nella comprensione delle funzionalità dell'applicazione, si è cercato di costruire un flusso di esecuzione ben strutturato ed intuitivo, in modo che venga naturale all'utente compiere le operazioni nell'ordine previsto. Per ottenere tale risultato si è cercato di avvicinare il più possibile il work-flow a quello delle azioni quotidiane dell'utente, imitando una situazione reale. Anche grazie all'ordine dei vari comandi nei menu, si è cercato di suggerire la sequenza di operazioni più funzionale possibile.

5.1.1. Casi d'uso

In questo paragrafo verranno analizzate le possibili operazioni che possono essere effettuate dall'utente attraverso la descrizione dei casi d'uso. Ogni caso d'uso rappresenta una specifica azione compiuta dall'utente e segnalata al sistema tramite l'apposito comando. Come si può notare dal diagramma in Figura 20, esistono due casi d'uso principali che l'utente può scegliere di utilizzare durante l'esecuzione: il primo è la scelta del modello tra le varie opzioni disponibili, che va effettuata come prima operazione in assoluto all'avvio dell'applicazione; la seconda è la dichiarazione di voler interagire con il modello stesso, grazie alla quale il sistema rende disponibili gli opportuni comandi.

Il caso d'uso "Interazione" è esteso da vari altri casi d'uso, che rappresentano le quattro possibili azioni che può fare l'utente per cambiare lo stato attuale del modello del divano, ovvero spostarlo nello spazio, ruotarlo, modificarne il colore (o, a seconda dei casi, il materiale) e rimuoverlo. Con la relazione "Extends" si vuole indicare un comportamento opzionale che estende, appunto, il normale flusso di esecuzione del caso d'uso che viene esteso: in questo caso, una volta iniziata l'interazione, l'utente può scegliere di eseguire una delle operazioni proposte o nessuna di esse.

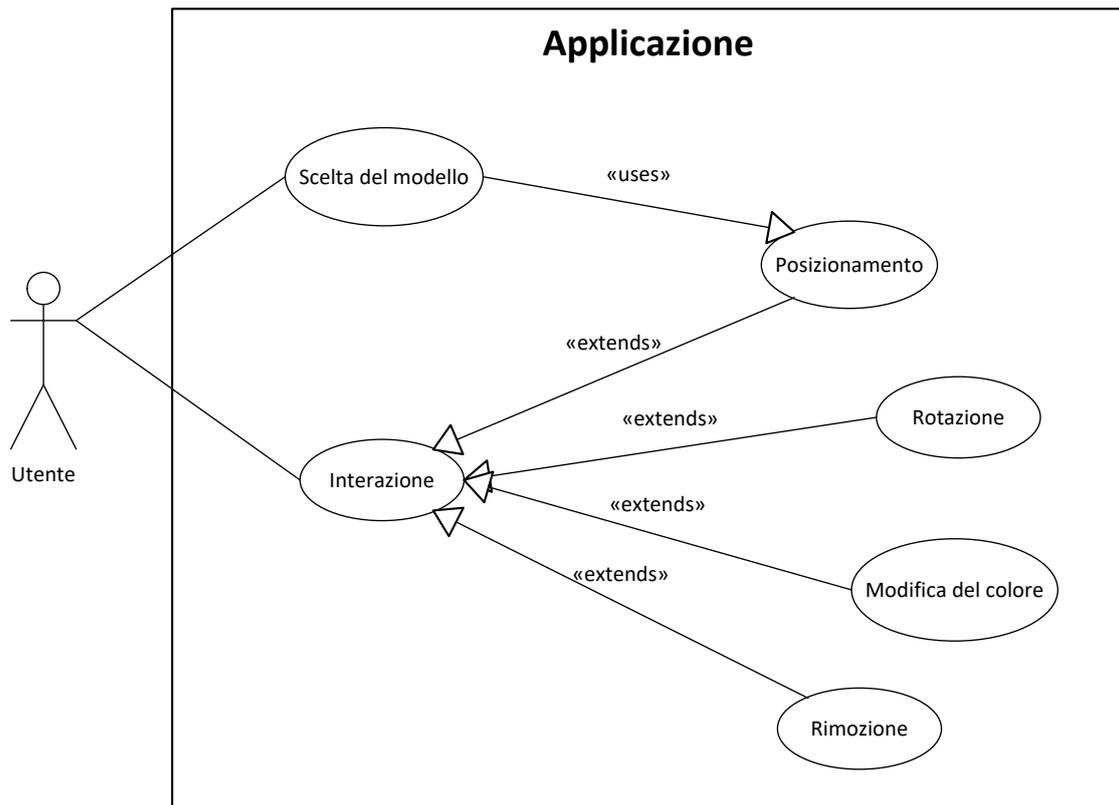


Figura 20 - Diagramma dei casi d'uso dell'applicazione

Si nota inoltre dal diagramma che “Scelta del modello” è legato al caso d’uso “Posizionamento” da una relazione di utilizzo, o di inclusione: dopo aver scelto il modello desiderato, infatti, viene attivata automaticamente l’operazione per il posizionamento nel divano nello spazio. Tale caso d’uso viene quindi utilizzato obbligatoriamente almeno una volta prima di poter iniziare l’interazione, ma può anche venire richiesto successivamente dall’utente in qualsiasi momento dell’utilizzo.

5.1.1.1. Descrizione dei casi d’uso

- **Scelta del modello:** per poter accedere a questo caso d’uso l’utente si deve trovare al primo passo dell’esecuzione, subito dopo l’avvio, oppure deve aver rimosso dalla scena il modello precedentemente selezionato. Se una di queste due condizioni è verificata, allora potrà procedere alla scelta del modello, selezionando quello desiderato tra le varie opzioni proposte. Una volta effettuata la scelta, l’applicazione renderà visibile il modello desiderato all’interno della scena.

- **Interazione:** l'utente può dichiarare di voler interagire con il modello se e solo se ha già effettuato l'operazione "Scelta del modello" e non ha ancora effettuato la "Rimozione", ovvero se il modello del divano è ancora visibile nella scena. Una volta segnalato al sistema di voler iniziare l'interazione, vengono presentati all'utente i comandi corrispondenti alle varie operazioni disponibili. A questo punto l'utente può selezionare una qualsiasi delle operazioni oppure può decidere di terminare l'interazione, provocando la scomparsa dei relativi comandi.
- **Posizionamento:** questo caso d'uso, come i successivi tre, rappresenta un'estensione del caso d'uso "Interazione"; l'utente deve quindi aver dato inizio all'interazione per poter accedere a questa funzionalità. Una volta dichiarato di voler riposizionare il divano, muovendolo dalla sua posizione attuale nello spazio, il sistema consente all'utente di decidere la nuova posizione del modello attraverso la direzione dello sguardo. Per terminare l'operazione l'utente deve confermare la posizione definitiva del divano, dopodiché lo stato di esecuzione sarà riportato a quello iniziale di questo caso d'uso, ovvero la scelta delle possibili operazioni di interazione.
- **Rotazione:** l'utente deve aver selezionato l'operazione di rotazione all'interno delle possibili opzioni di interazione. Il sistema presenta all'utente i comandi utili a effettuare l'operazione, ovvero quelli per la rotazione oraria, per la rotazione antioraria e per il salvataggio della posizione. Una volta raggiunto l'angolo di rotazione desiderato, attraverso il comando di salvataggio l'utente dichiara di voler memorizzare quella particolare posizione del modello e viene poi riportato allo stato di scelta dell'operazione di interazione.
- **Modifica del colore:** l'utente sceglie, tra i possibili tipi di interazione, di voler modificare il colore (o, se possibile per il modello selezionato, il materiale) del divano. Il sistema presenta all'utente le varie scelte e combinazioni disponibili per il modello correntemente visualizzato. Una volta che l'utente effettua la scelta, viene

apportata la modifica corrispondente al modello e l'esecuzione viene riportata allo stato di selezione dell'operazione.

- **Rimozione:** questo caso d'uso è leggermente diverso dai precedenti perché, sebbene anche qui l'utente debba aver dichiarato di voler interagire con il modello del divano e poi aver selezionato l'apposito comando, una volta terminata l'operazione lo stato dell'esecuzione verrà riportato a quello di partenza, ovvero al momento della selezione del modello. L'operazione vera e proprio consiste nella rimozione del divano dalla scena dell'applicazione; per proseguire sarà quindi necessario scegliere un nuovo modello con cui poi interagire.

5.1.1.2. Organizzazione della scena tridimensionale

Il primo passo per lo sviluppo dell'applicazione consiste nello strutturare la scena, scegliendo gli elementi opportuni ed inserendoli nella gerarchia come GameObject. Nella scena verranno inseriti tutti gli oggetti che potranno servire all'utente in un qualsiasi momento dell'esecuzione: quelli che all'avvio dell'applicazione non dovrebbero essere visibili saranno poi disattivati dalle impostazioni statiche di Unity o più avanti, tramite codice. La gerarchia della scena è stata organizzata in modo da avere tutti gli oggetti grafici, come i modelli dei divani o i pulsanti, come discendenti di un unico GameObject denominato "Divano", mentre gli altri tipi di elementi sono inseriti all'esterno di "Divano", come suoi fratelli. Oltre agli elementi contenuti in "Divano", i GameObject che costituiscono la scena della nostra applicazione sono Main Camera, Directional Light, Spatial Mapping, Cursor e Managers.

I componenti inseriti nel GameObject "Divano" sono quelli che caratterizzano fisicamente la scena (Figura 21): quelli che l'utente visualizza graficamente una volta avviata l'applicazione e con cui può interagire. Abbiamo quindi i modelli dei divani in tutte le possibili combinazioni di materiali e colori, in modo che in seguito ad ogni richiesta dell'utente possa essere attivato il modello corrispondente, ma anche i vari set di comandi che andranno utilizzati per le varie operazioni ed un testo tridimensionale introduttivo che guida l'utente nella prima operazione all'inizio dell'esecuzione.



Figura 21 - Scena tridimensionale dell'applicazione costruita in Unity

La Main Camera rappresenta il punto di vista dell'utente all'avvio dell'applicazione; viene posizionata nell'origine dello spazio virtuale in modo da facilitare la gestione della posizione di tutti gli altri ologrammi. Essendo un oggetto standard di Unity è sufficiente importare il "prefab" corrispondente nella gerarchia, con l'accortezza di impostare il colore di sfondo sul nero, che verrà interpretato in Windows Holographic come trasparente e consentirà di vedere l'ambiente reale attorno agli ologrammi. Anche nel caso di Directional Light viene importato un oggetto prefabbricato: si tratta di una luce direzionale di cui è possibile modificare le caratteristiche a piacimento, ad esempio cambiandone la direzione, l'intensità o il colore. Grazie all'inserimento di una fonte di luce nella scena, la resa degli ologrammi risulta più realistica e meglio integrata con l'ambiente reale; generalmente viene posizionata in alto e leggermente spostata dalla verticale degli ologrammi, in modo da simulare una generica fonte luminosa proveniente dal soffitto della stanza.

Spatial Mapping e Managers, invece, non hanno una rappresentazione grafica: sono dei GameObject vuoti a cui vengono associati degli specifici script. Spatial Mapping contiene il codice per la gestione del riconoscimento spaziale, suddiviso nei tre file di script "Spatial Mapping Observer", "Spatial Mapping Manager" e "File Surface Observer". Grazie alle funzioni definite in questi tre file è possibile interpretare la ricostruzione dello spazio reale effettuata dai sensori dell'HoloLens ed utilizzarla per far interagire gli ologrammi con lo spazio circostante. Analogamente, all'oggetto Managers sono associati i file di script per

l'interpretazione delle tre diverse tipologie di input: direzione dello sguardo, gesti e comandi vocali, suddivisi rispettivamente nei file "Gaze Manager", "Gesture Manager" e "Speech Manager".

Infine, l'oggetto Cursor rappresenta l'elemento grafico che evidenzia la posizione corrente del cursore, ovvero la direzione dello sguardo dell'utente. Il cursore che è stato utilizzato è costituito da due diversi componenti grafici, un cerchio blu ed una sferetta bianca, che si alternano durante l'utilizzo: se lo sguardo dell'utente è rivolto verso un ologramma e quindi il cursore è sovrapposto al modello dell'oggetto inquadrato, allora verrà utilizzato il cerchio, in caso contrario verrà mostrata la sfera. Le funzioni per la gestione del cursore sono implementate nei file "Cursor Manager" e "Basic Cursor".

5.1.2. Set di comandi per l'interazione

I pulsanti per i comandi di interazione sono stati realizzati graficamente come degli oggetti tridimensionali sferici, ognuno con un proprio simbolo che rimanda alla funzionalità corrispondente. I vari comandi sono divisi in quattro diversi set; non viene mai mostrato più di un set contemporaneamente ed è stato quindi possibile posizionarli tutti nel medesimo punto dello spazio, ovvero esattamente sulla verticale del divano. In questo modo, quando viene richiesta una particolare funzionalità, i comandi richiesti prendono il posto di quelli precedentemente visualizzati. Il set di comandi principale, da cui è possibile invocare le varie operazioni di interazione, è quello dei comandi di base ("Base Commands"); a partire da questo è possibile accedere ai set di comandi per la rotazione, per la scelta del modello e per la scelta del colore.

Ogni pulsante di ogni set è accompagnato da un testo tridimensionale che specifica il nome del comando, per chiarirne la funzione. Il testo non è sempre visibile ma viene attivato soltanto quando il cursore si posa sulla sfera di quello specifico bottone, operazione che precede necessariamente la selezione del comando. Inoltre, il testo mostrato corrisponde al comando vocale relativo alla stessa funzionalità, per aiutare l'utente ad associare ad ogni operazione la parola chiave corretta.

A tutti i pulsanti dei vari menu, come anche a tutti i testi ad essi associati, è stato applicato il file di script “Billboard”, per fare in modo che sia i simboli che le scritte siano sempre orientate nella direzione dell’utente, in modo che i menu possano essere utilizzati da qualsiasi posizione. Per ottenere tale comportamento viene utilizzata un’unica funzione che, ad ogni aggiornamento del display, ottiene un vettore tridimensionale che rappresenta la direzione dello sguardo dell’utente e lo utilizza per modificare la rotazione dell’oggetto selezionato ed orientarlo verso l’utente (Script 1).

```
/// <summary>
/// Keeps the object facing the camera.
/// </summary>
private void Update()
{
    // Get a Vector that points from the Camera to the target.
    Vector3 directionToTarget = Camera.main.transform.position -
        gameObject.transform.position;

    // If we are right next to the camera the rotation is undefined.
    if (directionToTarget.sqrMagnitude < Mathf.Epsilon)
    {
        return;
    }

    // Calculate and apply the rotation required to reorient the object and
    // apply the default rotation to the result.
    gameObject.transform.rotation = Quaternion.LookRotation(-directionToTarget)
        * DefaultRotation;
}
```

Script 1 – Codice tratto da billboard.cs per la rotazione degli oggetti verso la posizione dell’utente

5.1.2.1. Base Commands

Il set “Base Commands” è costituito da quattro pulsanti, che rappresentano le quattro operazioni di interazione analizzate precedentemente nella descrizione dei casi d’uso: spostamento, rotazione, modifica del colore e rimozione (Figura 22). A questi quattro comandi sono state associate rispettivamente le quattro parole chiavi “Move”, “Rotate”, “Settings” e “Remove”. L’esecuzione di queste quattro funzionalità viene affrontata in maniera differente a seconda dei casi: in alcuni il “tap” provoca la comparsa di un nuovo

menu, specifico per l'operazione selezionata, in altri invece l'azione viene eseguita immediatamente.



Figura 22 - Comandi di base per l'interazione

Nel caso dello spostamento viene ad esempio attivata direttamente la funzionalità richiesta, e l'utente può riposizionare il divano nello spazio utilizzando la direzione dello sguardo; nel caso di rotazione e modifica del colore, invece, vengono visualizzati gli appositi menu, rispettivamente "Rotation Commands" e "Color Choice", che sostituiscono i comandi precedenti. Nel quarto e ultimo caso, invece, viene immediatamente effettuata l'operazione, ovvero la rimozione del divano dallo spazio, e inoltre viene mostrato il menu per la selezione del modello ("Sofa Choice"), per consentire all'utente di iniziare una nuova interazione.

5.1.2.2. Rotation Commands

Il set di comandi per effettuare la rotazione del modello viene mostrato automaticamente quando viene selezionato il comando "Rotate" del menu di base. Questo set è costituito da tre semplici comandi, realizzati anch'essi come pulsanti sferici: "Left", "Right" e "Ok", rispettivamente per la rotazione antioraria, per la rotazione oraria e per il salvataggio della posizione (Figura 23). Una volta attivata l'operazione di rotazione l'utente potrà quindi cambiare l'angolazione del divano a piacimento finché non sarà soddisfatto della posizione raggiunta; a quel punto termina l'operazione selezionando il pulsante "Ok" e il menu di rotazione viene sostituito da quello dei comandi di base, per permettere altre eventuali interazioni.

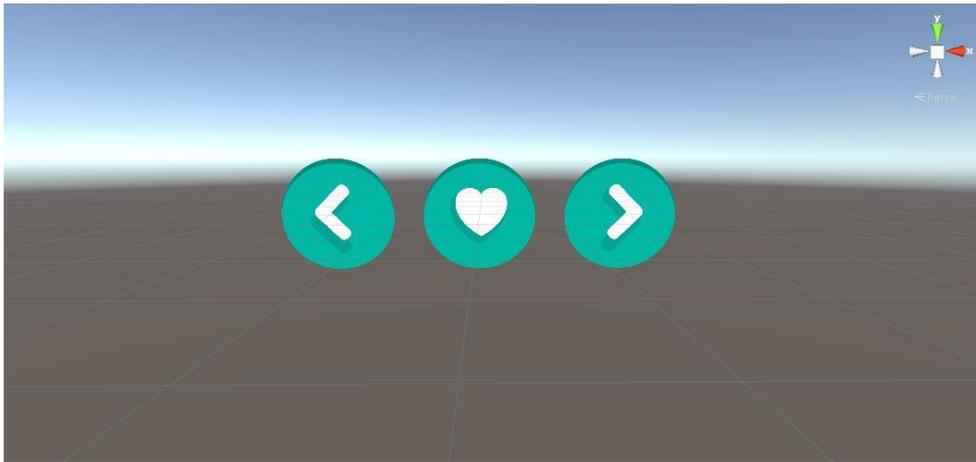


Figura 23 - Comandi per la rotazione

5.1.2.3. Sofa Choice e Color Choice

I due menu per la scelta del modello (Figura 24) e per la scelta del colore (Figura 25) hanno strutture analoghe, sebbene siano utilizzati in momenti differenti dell'esecuzione. Il primo viene utilizzato all'avvio dell'applicazione ed ogni volta che viene rimosso il divano dallo spazio, ovvero quando deve essere scelto uno dei modelli disponibili per poi iniziare l'interazione; il secondo, invece, costituisce una delle possibili operazioni di interazione e viene mostrato quando l'utente seleziona il relativo comando del menu di base.

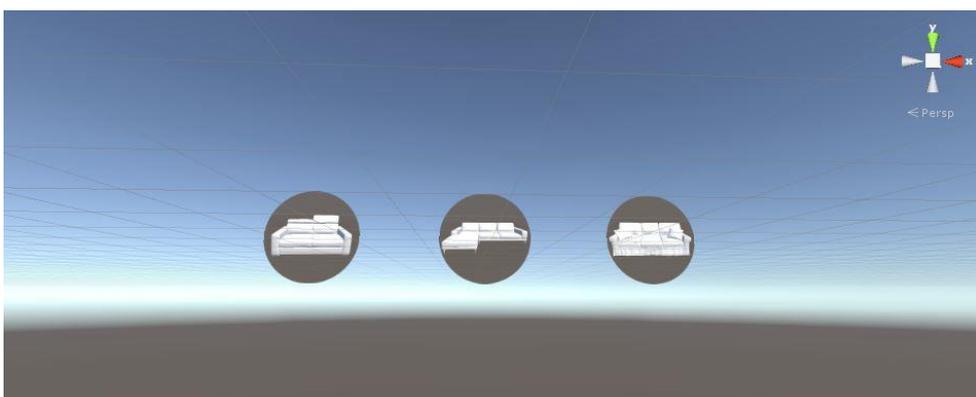


Figura 24 - Comandi per la scelta del modello

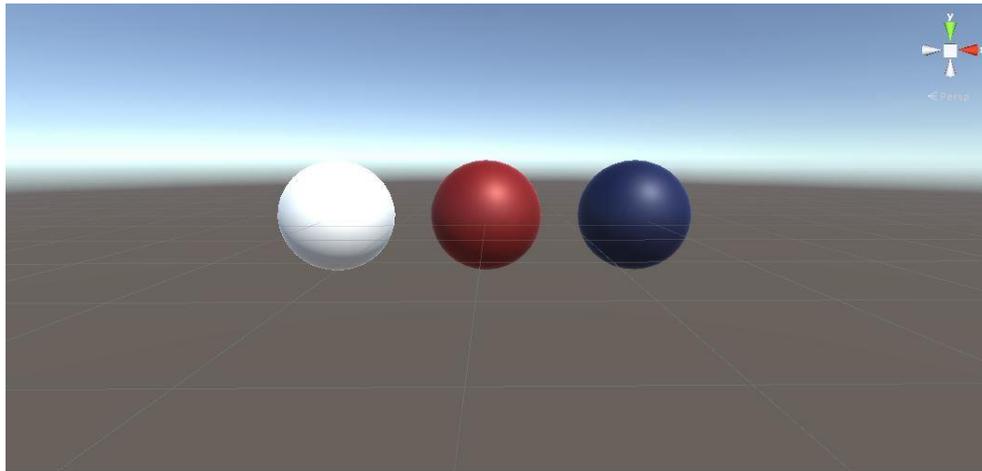


Figura 25 - Comandi per la scelta del colore

In entrambi i casi, abbiamo dei pulsanti sferici che rappresentano tutte le opzioni disponibili, sia che si tratti di modelli di divani differenti che di possibili colori per lo stesso modello. Selezionando uno dei pulsanti la scelta dell'utente viene applicata alla scena, mostrando il divano richiesto. Una volta effettuata l'operazione, viene attivato il menu di base per consentire all'utente di effettuare nuove operazioni di interazione.

5.2. Gestione dei dati di input

La realizzazione dell'interazione dell'utente con gli ologrammi è stata effettuata tramite la gestione delle tre tipologie di input previste dall'HoloLens: sguardo, gesti e voce. I primi due sono stati utilizzati insieme per realizzare i vari menu con pulsanti interattivi; l'ultimo, invece, è servito per la realizzazione dei comandi vocali, indipendenti ed alternativi a quelli gestuali.

5.2.1. Managers

Il codice per la gestione dei vari tipi di input è stato suddiviso in tre diversi file di script, ognuno relativo ad una delle tre tipologie del gruppo GGv (gaze, gesture, voice): abbiamo infatti "Gaze Manager" per l'interpretazione della direzione dello sguardo, "Gesture Manager" per l'elaborazione dei gesti dell'utente e "Speech Manager" per i comandi vocali.

Questi tre file sono stati associati ad un unico GameObject, denominato “Managers”, che non contiene componenti grafici ma il cui unico scopo è di fare da contenitore per i file di script in modo da poterli inserire nella gerarchia della scena.

5.2.2. Gaze Manager

Il compito del Gaze Manager è di gestire le informazioni che riguardano lo sguardo dell'utente. È particolarmente importante conoscere in ogni momento la direzione dello sguardo per poterlo utilizzare come puntatore; a seconda dell'ologramma su cui si trova il cursore, uno stesso gesto può causare comportamenti differenti. Ad ogni aggiornamento del display, il Gaze Manager calcola tali informazioni tramite la funzione “Physics.Raycast” e, nel caso in cui lo sguardo dell'utente stia puntando un ologramma, dopo aver salvato un riferimento all'ultimo oggetto puntato, memorizza tale ologramma come “FocusedObject”; in caso contrario, “FocusedObject” viene settato a “null”.

Successivamente, Gaze Manager confronta l'oggetto correntemente puntato dall'utente con quello salvato in memoria, ovvero l'ologramma che risultava puntato al precedente aggiornamento del frame. Se i due oggetti non corrispondono, viene segnalato il cambiamento ad entrambi, attraverso due diversi messaggi: “OnGazeLeave” viene inviato all'oggetto da cui lo sguardo dell'utente è appena uscito, “OnGazeEnter” è inviato all'oggetto che l'utente ha appena iniziato a guardare (Script 2).

Il Gaze Manager è alla base di gran parte delle operazioni di interazione: le proprietà esposte vengono utilizzate dal Gesture Manager, come vedremo nel paragrafo successivo, per ottenere informazioni sull'oggetto correntemente puntato ad applicare i comandi gestuali a quel preciso oggetto. Le funzionalità del Gaze Manager possono però anche utilizzate indipendentemente quando si vogliono ottenere comportamenti legati unicamente alla posizione dello sguardo: è il caso delle didascalie presenti sotto i vari pulsanti, che vengono mostrate e nascoste in seguito alla ricezione dei messaggi “OnGazeEnter” e “OnGazeLeave”.

```

/// <summary>
/// Calculates the Raycast hit position and normal.
/// </summary>
private void UpdateRaycast()
{
    // Get the raycast hit information from Unity's physics system.
    RaycastHit hitInfo;
    Hit = Physics.Raycast(gazeOrigin,
        gazeDirection,
        out hitInfo,
        MaxGazeDistance,
        RaycastLayerMask);

    GameObject oldFocusedObject = FocusedObject;
    // Update the HitInfo property so other classes can use this hit information.
    HitInfo = hitInfo;

    if (Hit)
    {
        // If the raycast hits a hologram, set the position and normal to match the
        //intersection point.
        Position = hitInfo.point;
        Normal = hitInfo.normal;
        lastHitDistance = hitInfo.distance;
        FocusedObject = hitInfo.collider.gameObject;
    }
    else
    {
        // If the raycast does not hit a hologram, default the position
        // to last hit distance in front of the user,
        // and the normal to face the user.
        Position = gazeOrigin + (gazeDirection * lastHitDistance);
        Normal = -gazeDirection;
        FocusedObject = null;
    }

    // Check if the currently hit object has changed
    if (oldFocusedObject != FocusedObject)
    {
        if (oldFocusedObject != null)
        {
            oldFocusedObject.SendMessage("OnGazeLeave",
                SendMessageOptions.DontRequireReceiver);
        }
        if (FocusedObject != null)
        {
            FocusedObject.SendMessage("OnGazeEnter",
                SendMessageOptions.DontRequireReceiver);
        }
    }
}

```

Script 2 – Codice per la generazione degli eventi OnGazeEnter e OnGazeLeave

5.2.3. Gesture Manager

Il compito del Gesture Manager è di effettuare il riconoscimento dei comandi gestuali; dei due possibili comandi disponibili in Windows Holographic, uno dei due ha un funzionamento standard ed è utilizzato per terminare l'esecuzione dell'applicazione. Il riconoscimento del gesto di "bloom" è infatti sempre attivo e scatena sempre lo stesso evento. Il gesto di "tap", invece, può essere gestito esplicitamente per realizzare le funzionalità desiderate. Il riconoscimento dei gesti viene realizzato tramite un oggetto "GestureRecognizer", che si occupa di interpretare le immagini ottenute dalla fotocamera dell'HoloLens e di scatenare opportuni eventi quando riconosce uno dei comandi gestuali previsti.

In Gesture Manager, all'interno della funzione "Start" di inizializzazione viene agganciato all'evento corrispondente al "tap" un'apposita funzione; tale funzione, denominata "OnTap", realizza la comunicazione dell'evento all'oggetto interessato. Il riferimento all'oggetto con cui l'utente sta interagendo viene ottenuto tramite il Gaze Manager, che come visto precedentemente restituisce un riferimento all'ologramma verso cui è rivolto lo sguardo dell'utente. A questo punto viene inviato all'oggetto in questione un messaggio "OnSelect", che sarà utilizzato per realizzare i comportamenti desiderati tramite un'opportuna gestione dell'evento (Script 3).

```
void Start()
{
    // Create a new GestureRecognizer. Sign up for tapped events.
    gestureRecognizer = new GestureRecognizer();
    gestureRecognizer.SetRecognizableGestures(GestureSettings.Tap);

    gestureRecognizer.TappedEvent += GestureRecognizer_TappedEvent;

    // Start looking for gestures.
    gestureRecognizer.StartCapturingGestures();
}
private void OnTap()
{
    if (focusedObject != null)
    {
        focusedObject.SendMessage("OnSelect");
    }
}
private void GestureRecognizer_TappedEvent(InteractionSourceKind source,
    int tapCount, Ray headRay)
{
    OnTap();
}
```

Script 3 – Codice per la gestione del comando gestuale Air Tap

5.2.4. Speech Manager

Per effettuare il riconoscimento dei comandi vocali il primo passo è inizializzare un dizionario in cui andranno inserite tutte le parole chiavi che dovranno essere riconosciute come comandi vocali; ad ogni parola chiave andrà associata una “lambda expression”, ovvero una funzione anonima passabile come parametro che descriva il comportamento desiderato. Il dizionario dovrà poi essere utilizzato a sua volta per inizializzare un oggetto di tipo “KeywordRecognizer”, il cui compito sarà quello di scatenare opportuni eventi in seguito al riconoscimento di una delle parole chiave specificate.

Nella funzione “Start” dello Speech Manager, dopo aver effettuato le inizializzazioni necessarie, va agganciato all’evento di riconoscimento di una parola chiave un’apposita funzione di gestione; deve poi essere invocata la funzione “Start” del KeywordRecognizer per avviare il riconoscimento vocale. Nella funzione di gestione dell’evento andrà poi effettuata l’invocazione della lambda, in modo che quando l’utente pronunci una delle parole chiave presenti nel dizionario venga automaticamente eseguita la funzione corrispondente (Script 4).

```
private void KeywordRecognizer_OnPhraseRecognized(PhraseRecognizedEventArgs args)
{
    System.Action keywordAction;
    if (keywords.TryGetValue(args.text, out keywordAction))
    {
        keywordAction.Invoke();
    }
}
```

Script 4 – Codice per l’esecuzione di un’azione in seguito al riconoscimento di un comando vocale

5.3. *Operazioni di interazione*

Le quattro principali operazioni di interazione con il modello possono essere eseguite dall'utente in qualsiasi ordine e possono essere ripetute infinite volte; proprio come se l'utente stesse spostando e ruotando un divano concreto nel proprio salotto.

5.3.1. Memorizzazione dello stato attuale

Per salvare in memoria lo stato corrente dell'esecuzione è stata utilizzata la proprietà "Tag" del GameObject "Divano": il Tag è costituito da una stringa di testo che può assumere un valore qualunque tra quelli predefiniti, modificabile dinamicamente durante l'esecuzione. Da Unity è possibile accedere e modificare l'elenco dei Tag disponibili: sono state quindi aggiunte delle stringhe adatte a rappresentare i vari stati dell'esecuzione dell'applicazione, ovvero la scelta del modello, la rotazione, il posizionamento e la scelta del colore.

A seconda dello stato corrente, ricavabile in ogni momento dalla proprietà Tag, vengono mostrati e nascosti opportunamente i vari set di comandi e abilitate o disabilitate le opportune funzionalità. Ad esempio, quando viene richiesta l'operazione di posizionamento il Tag viene settato a "Placing" e viene resa visibile la mappatura dello spazio tramite un apposito reticolato bianco, per aiutare l'utente ad individuare le zone dove poter posizionare il divano. Analogamente, ad ogni stato dell'esecuzione corrisponde una particolare configurazione di funzionalità, per fare in modo che si possa visualizzare in ogni momento solo gli elementi effettivamente utili e che venga seguito il flusso di esecuzione previsto.

La gestione dello stato tramite Tag è inoltre fondamentale nell'elaborazione dei comandi vocali: non tutti i comandi possono essere eseguiti in ogni momento, ma attraverso la lettura dello stato corrente viene stabilito se eseguire effettivamente l'azione richiesta o se ignorare il comando in caso di una funzionalità non prevista per quello stato. In alcuni casi, ad una stessa parola chiave possono essere associati diversi comportamenti, che variano a seconda della situazione attuale: pronunciando "Ok", ad esempio, si può fermare il divano in un punto se lo si stava posizionando, oppure salvarne l'angolazione se lo si stava ruotando. Generalmente si è evitato di associare molteplici comportamenti ad un unico comando

vocale, ma in alcuni casi è stata ritenuta una scelta opportuna per migliorare l'intuitività dell'applicazione.

5.3.2. Selezione del modello ed avvio dell'interazione

All'avvio dell'applicazione vengono presentati all'utente i vari modelli disponibili, sotto forma di oggetti sferici (Figura 26). Su ogni sfera è rappresentata l'immagine bidimensionale del divano corrispondente; l'utente dovrà spostare lo sguardo fino a che il cursore non sarà posizionato su una delle sfere, e poi selezionarlo utilizzando il comando gestuale "Tap". Nel momento in cui il cursore è posizionato su uno dei pulsanti, la classe Gaze Manager invia al GameObject corrispondente un messaggio di tipo "OnGazeEnter", per segnalare che l'utente ha appena iniziato a fissare tale oggetto. Tramite un'apposita funzione di gestione dell'evento, viene quindi mostrato un testo didascalico posizionato subito sotto la sfera, che specifica il nome del modello correntemente puntato. Lo stesso testo verrà analogamente nascosto non appena l'utente muoverà lo sguardo verso un altro ologramma, grazie alla gestione dell'evento opposto "OnGazeLeave".

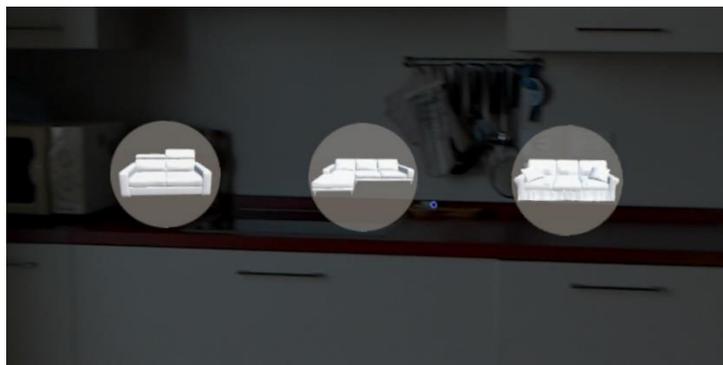


Figura 26 - Menu per la selezione del modello

In seguito alla selezione di uno dei divani, l'applicazione nasconderà i pulsanti sopra descritti, ormai non più necessari, e mostrerà invece il modello selezionato. Una volta scelto il modello viene invocata automaticamente l'operazione di posizionamento; in questo modo il divano non viene posizionato casualmente nella stanza, ma viene lasciato all'utente il compito di selezionare un luogo adatto, attraverso gli specifici comandi illustrati nel

paragrafo successivo. Sia il punto in cui si trova il divano che la sua angolazione rispetto alla posizione dell'utente potranno successivamente essere modificati a piacimento. Per iniziare a interagire con il modello è sufficiente posizionare il cursore su un qualunque punto del divano ed eseguire il gesto "Tap": in tal modo verrà mostrato un nuovo menu, quello denominato "Base Commands". Tale menu è costituito da quattro diversi pulsanti sferici, che danno accesso alle quattro principali funzionalità di interazione: posizionamento, rotazione, modifica del colore e rimozione. I pulsanti potranno essere nascosti in qualsiasi momento eseguendo nuovamente il "Tap" sul divano, per visualizzare più comodamente il modello e valutarne l'effetto finale.

5.3.3. Posizionamento del modello nello spazio reale

La prima operazione che viene presentata all'utente nel menu dei comandi per l'interazione è il posizionamento; è stata inserita in prima posizione poiché generalmente è la prima operazione effettuata da un utente che voglia valutare l'effetto di un particolare divano in una stanza ed è rappresentata dal simbolo di un fulmine per evocare l'idea di movimento. Il posizionamento avviene grazie all'utilizzo combinato dei gesti e della direzione dello sguardo: per dare inizio al posizionamento va selezionato il pulsante relativo tramite "Tap"; a questo punto lo stato dell'applicazione viene impostato su "Placing": ciò vuol dire che finché si permane in questo stato il divano non è ancorato in modo fisso allo spazio reale ma è libero di muoversi, ovvero di seguire la direzione dello sguardo dell'utente.

L'utilizzatore dovrà quindi ruotare la testa, anche mentre cammina liberamente nella stanza, finché non inquadra il punto esatto in cui desidera posizionare il divano. Per segnalare la fine dell'operazione è sufficiente eseguire nuovamente il "Tap": il divano sarà fissato nel punto richiesto e l'esecuzione uscirà dallo stato di posizionamento. Per tutta durata dell'operazione, dalla selezione del comando alla conferma della posizione, i comandi di base saranno nascosti e verrà invece visualizzato un reticolato bianco che ricalchi il modello tridimensionale della stanza così come è stato memorizzato dall'HoloLens (Figura 27). In tal modo l'utente potrà individuare più facilmente le zone dello spazio adatte ad ospitare il modello del divano, ma anche accorgersi di eventuali errori nel mapping, dovuti principalmente a oggetti o persone in movimento; tali errori verranno

corretti automaticamente durante l'esecuzione grazie all'azione continua in background dello Spatial Mapping Manager.



Figura 27 - Posizionamento del modello nello spazio

5.3.4. Rotazione del modello

Una volta che il modello è stato posizionato nel punto desiderato, l'operazione che risulta utile effettuare è la rotazione, per trovare la giusta angolazione del divano rispetto allo spazio circostante. Per questo, il secondo pulsante del menu per l'interazione è proprio quello relativo alla rotazione, rappresentata graficamente da una freccia circolare. A differenza del comando precedente, in questo caso l'interazione avviene solamente tramite il comando gestuale "Tap": perciò, quando viene richiesta l'operazione di rotazione, il menu di base viene nascosto e vengono mostrati tre nuovi pulsanti che consentono di modificare l'angolazione del modello (Figura 28).



Figura 28 – Esempio di rotazione del modello

Durante la rotazione del divano l'applicazione si trova nello stato denominato "Rotating", in cui l'utente non può accedere alle altre operazioni di interazione ma può soltanto interagire con il modello tramite i tre pulsanti visualizzati. Dopo aver raggiunto l'angolazione desiderata utilizzando i due pulsanti per la rotazione oraria ed antioraria, caratterizzati da due frecce con versi opposti, l'utente segnala di voler salvare la posizione corrente tramite il terzo e ultimo pulsante. Il bottone per il salvataggio riprende la simbologia comune sul web, e in particolare sui social network, secondo cui il simbolo del cuore è associato all'idea di apprezzamento ("like"), in questo caso nei confronti dell'angolazione raggiunta. Una volta scelta l'angolazione definitiva, l'applicazione esce dallo stato di rotazione e viene mostrato nuovamente il menu dei comandi di interazione.

5.3.5. Modifica del colore del modello

Dopo aver scelto la posizione e l'angolo di rotazione desiderati, l'utente può modificare il colore del divano per valutarne l'effetto in relazione all'ambiente e scegliere quello che più lo aggrada. Per semplicità sono stati implementati solo alcuni campioni tratti dalla vasta gamma di colori offerta da Polotronesofà: è possibile scegliere tra tre colori piatti, applicabili a tutti i modelli di divano presenti nell'applicazione, e volutamente privi di pattern e di fantasie elaborate in modo da poter corrispondere ai gusti di un maggior numero di utenti.



Figura 29 – Scelta del colore del modello

Essendo i colori proposti applicabili a tutti i diversi divani, il menu di selezione per la scelta del colore è unico: è stato realizzato, analogamente ai menu precedenti, tramite pulsanti sferici e viene invocato quanto si clicca sul terzo pulsante del menu per l'interazione. La figura scelta per questo particolare comando è un ingranaggio, simbolo spesso collegato all'idea di personalizzazione e di modifica delle proprietà. Una volta selezionato il comando, il menu di base viene nascosto e viene sostituito dal set di comandi "Color Choice", in cui ad ogni pulsante sferico corrisponde uno dei colori disponibili (Figura 29). Per realizzare questi pulsanti sono stati utilizzati dei semplici modelli tridimensionali sferici a cui è stato applicato, in Unity, esattamente lo stesso materiale utilizzato per i modelli dei divani; in questo modo l'utente ha un'idea esatta dell'effetto dei vari colori ancora prima di applicarli al divano. Quando viene selezionata una delle opzioni proposte l'azione termina automaticamente: la modifica richiesta viene applicata al modello, i pulsanti per la scelta del colore vengono nascosti e si ritorna alla modalità di interazione, per consentire eventuali nuove operazioni.

5.3.6. Rimozione del modello dallo spazio

Dopo aver utilizzato i tre comandi precedenti per scegliere la posizione, l'angolazione ed il colore desiderati, l'utente può voler terminare l'operazione corrente per provare a visualizzare un modello differente. Per questo motivo viene offerta l'operazione di rimozione, invocabile selezionando l'ultimo dei quattro pulsanti del menu (Figura 30). Rimuovendo il modello del divano dallo spazio, l'utente è libero di cominciare una nuova interazione, riiniziando dalla scelta del modello tra quelli disponibili, senza essere costretto a dover riavviare l'applicazione.



Figura 30 - Comandi per le operazioni di interazione

Quando è selezionato il comando di rimozione tutte le precedenti personalizzazioni del modello vengono azzerate, tranne la posizione: l'angolazione torna ad essere frontale rispetto all'utente ed il colore quello di default. Si è scelto di mantenere la posizione corrente poiché durante l'utilizzo l'utente potrebbe essersi spostato in modo notevole nella stanza, o potrebbe essersi voltato in un'altra direzione: se si scegliesse di resettare la posizione l'utente potrebbe avere difficoltà nel ritrovare gli ologrammi nello spazio. Utilizzando la posizione corrente, invece, i comandi appariranno esattamente nella direzione del suo sguardo. Ad ogni modo la posizione corrente viene mantenuta soltanto in un primo momento, poiché iniziando una nuova interazione verrà richiesto all'utente di scegliere la posizione che ritiene più adatta all'interazione con il nuovo divano. Una volta rimosso il

modello corrente, viene automaticamente mostrato il menu “Sofa Choice”, per consentire all’utente di selezionare un nuovo modello con cui interagire. Nel caso in cui invece si voglia definitivamente terminare l’esecuzione dell’applicazione, sarà necessario eseguire il gesto di “bloom”.

5.3.7. Comandi vocali

Ognuno dei comandi illustrati in questa sezione è invocabile tramite un apposito comando vocale, utilizzabile alternativamente ai comandi gestuali. Le parole chiave che vengono riconosciute dall’HoloLens sono quelle indicate nella didascalia di ogni pulsante nei vari menu e sono in lingua inglese; le didascalie vengono mostrate all’utente non appena egli posa lo sguardo su uno dei pulsanti. Grazie ai comandi vocali è possibile utilizzare l’applicazione in tutta la sua potenzialità con il solo ausilio della voce; l’unico vincolo è quello di trovarsi in un ambiente sufficientemente silenzioso da permettere il riconoscimento vocale, condizione generalmente rispettata dato che si prevede l’utilizzo dell’applicazione in luoghi privati e di dimensioni limitate.

Naturalmente non tutti i comandi vocali sono disponibili in ogni momento dell’esecuzione, ma il riconoscimento di un comando avviene soltanto quando sono verificate le condizioni per l’esecuzione dell’azione corrispondente. All’avvio dell’applicazione, ovvero nello stato “Choosing”, gli unici comandi consentiti sono quelli di selezione del modello; una volta scelto il divano si possono utilizzare i comandi vocali per l’invocazione delle quattro operazioni di interazione. In modalità “Placing”, si può effettuare esclusivamente l’operazione di conferma della posizione, in modalità “Rotating” si può ruotare il divano nei due sensi e salvarne l’angolazione, in modalità “Changing”, infine, si può selezionare il colore del modello tra quelli proposti.

6. RISULTATI OTTENUTI E PERFORMANCE

6.1. Flusso di esecuzione dell'applicazione

L'utilizzo dell'applicazione finale risulta scorrevole ed intuitivo: trattandosi di un'applicazione con un solo scopo principale, si è cercato di facilitare il più possibile il percorso dell'utente accompagnandolo e supportandolo nei vari passi dell'esecuzione. Per ottenere tale risultato si è deciso di adottare una struttura a stati in cui l'utente non ha accesso contemporaneamente a tutte le funzionalità disponibili, ma in ogni stato può utilizzare solamente quelle previste; in questo senso è stata particolarmente utile la suddivisione dei vari comandi in menu differenti, in modo da poterli mostrare e nascondere all'occorrenza.

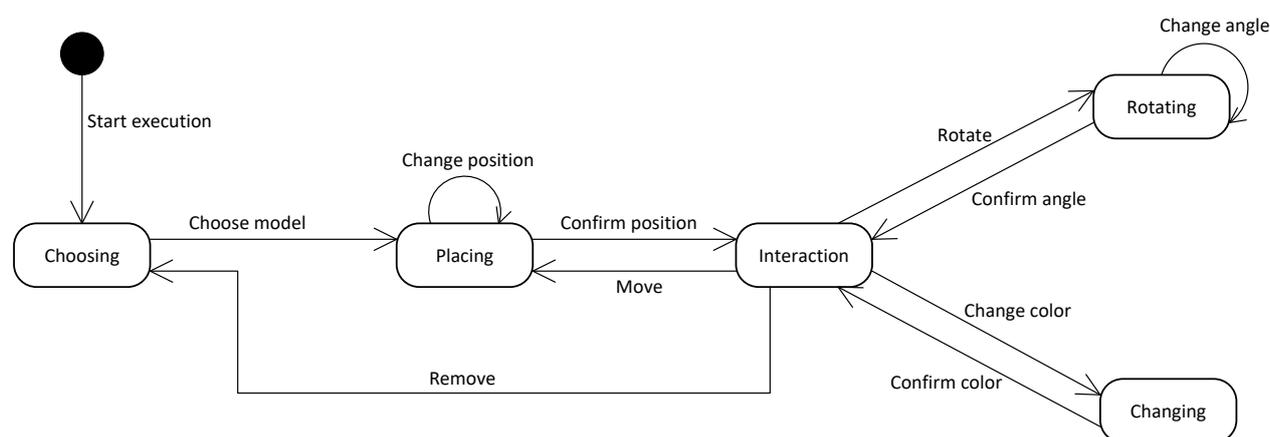


Figura 31 - Diagramma di stato dell'applicazione

Come si vede in Figura 31, all'avvio dell'esecuzione l'utente ha un'unica possibilità, ovvero la scelta del modello; una volta effettuato questo passaggio l'esecuzione passerà allo stato di "Placing", per la scelta della posizione in cui far comparire l'ologramma del divano, per poi arrivare allo stato "Interaction", a partire dal quale si ha la possibilità di effettuare una qualsiasi delle operazioni di interazione. La sequenza prevista per le varie operazioni sarebbe quella di posizionamento, rotazione ed infine scelta del colore, che è stata valutata come l'ordine di utilizzo delle funzionalità più logico per un utente che voglia provare a inserire il divano nella stanza e a valutarne l'effetto; i pulsanti del menu di interazione sono

stati infatti posizionati seguendo tale ordine, in modo da suggerirlo indirettamente all'utilizzatore.

Ciò non toglie che l'utente sia libero di scegliere l'ordine che ritiene più consono, o di eseguire solo alcune delle operazioni disponibili, o anche di utilizzare più volte una stessa funzionalità. L'utente viene infatti riportato allo stato "Interaction", da cui può richiedere le varie operazioni; potrebbe ad esempio essere necessario ripetere più volte il posizionamento perché l'utente sia soddisfatto del risultato raggiunto, oppure cambiare ripetutamente il colore per confrontare le varie opzioni proposte. Al contrario, l'utente potrebbe essere soddisfatto dopo aver compiuto solo alcune delle possibili azioni, ed esempio se è già soddisfatto dal colore di default del modello.

L'unico caso in cui l'esecuzione non segue il ciclo appena descritto è quello dell'operazione "Remove"; una volta rimosso il divano dallo spazio, infatti, l'utente viene riportato nella condizione iniziale "Choosing", ovvero di selezione del modello. In questo modo è possibile iniziare una nuova sessione di interazione, proprio come se l'applicazione fosse stata riavviata: angolazione e colore tornano ad essere quelli default e soltanto la posizione corrente viene conservata per essere utilizzata nella visualizzazione del menu iniziale. In generale, si è cercato di rendere la sessione di interazione il più realistica possibile, prendendo spunto da una situazione di vita reale in cui l'utente, dopo aver portato a casa il divano appena acquistato, ne cerchi la posizione migliore all'interno del salotto.

6.2. *Test di utilizzo*

Lo sviluppo dell'applicazione è stato affiancato da una continua attività di test, già dalle prime fasi della realizzazione. In un primo momento si è preso in considerazione l'utilizzo dell'emulatore offerto da Microsoft per simulare il funzionamento dell'HoloLens senza dover ricorrere al dispositivo fisico; dopo alcune prove, però, è stato ritenuto più pratico l'utilizzo del dispositivo reale sia a causa dell'eccessivo carico computazionale richiesto dall'emulatore, che rallentava le operazioni di test, sia per l'incompletezza dell'esperienza fornita dall'emulatore. HoloLens Emulator, infatti, replica in modo esatto quelli che sono gli elementi virtuali che verranno presentati all'utente, ma naturalmente essendo privo di tutti

i sensori presenti nel visore reale manca la parte di interazione tra gli ologrammi e lo spazio circostante. Avendo la possibilità di accedere ad un visore concreto, è stato scelto di utilizzarlo già dalle prime fasi di test, velocizzando ulteriormente lo sviluppo grazie alla comodità del deployment su macchina remota, che consente di mettere in esecuzione l'applicazione sul dispositivo direttamente da Visual Studio tramite rete wireless.

Le attività di test hanno avuto un peso considerevole nella realizzazione dell'applicazione; questo perché, avendo a che fare per la prima volta con una tecnologia innovativa e con un dispositivo hardware poco familiare, è stato necessario testare non solo le parti dell'applicazione sviluppate in proprio, ma anche i vari servizi offerti dalla piattaforma, in modo da poterne valutare i limiti e le potenzialità. Nelle prime fasi dello sviluppo, infatti, sono state realizzate piccole applicazioni di esempio che utilizzassero servizi come lo Spatial Mapping, lo Spatial Sound o il riconoscimento vocale, per analizzarne il funzionamento e poter decidere quali servizi utilizzare e come servirsene all'interno dell'applicazione.

I test svolti sono serviti quindi sia ad analizzare e migliorare le caratteristiche dell'applicazione che a valutare le possibilità di utilizzo del visore HoloLens. Per quanto riguarda l'applicazione realizzata, attraverso i test effettuati si è cercato di renderne l'utilizzo semplice ed intuitivo, con l'obiettivo di avvicinare il più possibile l'esperienza di realtà mista a quella della vita reale; infatti, più le funzionalità ed i gesti sono vicini all'esperienza quotidiana, più è facile per gli utenti interfacciarsi con l'applicazione. Dal punto di vista grafico, invece, sono stati testati diversi modelli 3D per osservarne la resa: pur avendo l'HoloLens un'elevata "densità olografica", ovvero un alto numero di punti luminosi utilizzati per rappresentare un ologramma, Unity impone un limite sul numero massimo di vertici e di poligoni che compongono un elemento, obbligando lo sviluppatore ad utilizzare modelli non particolarmente dettagliati. Tale vincolo può essere superato dividendo il modello che si desidera utilizzare in più elementi, in modo che le singole parti rispettino il limite numerico, ma tale procedimento richiede un lavoro di elaborazione del modello 3D a monte dello sviluppo dell'applicazione.

Per quanto riguarda il test delle funzionalità offerte dell'HoloLens, sono stati riscontrati alcune limitazione intrinseche del visore: la prima, e forse la più importante, riguarda il campo di azione degli ologrammi. Infatti, sebbene il visore sia completamente trasparente e consenta di avere una visuale quasi completa dello spazio circostante, i display su cui

vengono proiettati gli elementi virtuali ricoprono soltanto una zona ridotta, posta al centro del campo visivo. Per questo motivo spesso si ha l'effetto sgradevole di un taglio netto dell'ologramma quando esso esce dalla zona centrale, annullando il realismo dell'esperienza dell'utente; per la stessa ragione si è costretti a mantenere una certa distanza dagli ologrammi, in modo che possano entrare interamente nel campo visivo, soprattutto nel caso di oggetti di dimensioni notevoli come i divani della nostra applicazione.

Un'altra limitazione riscontrata nell'utilizzo del visore riguarda il riconoscimento gestuale: sebbene i comandi vengano riconosciuti con grande precisione, anche quando vengono eseguiti velocemente o con delle incertezze nei movimenti, il campo di azione in cui il riconoscimento è attivo è piuttosto ridotto. Probabilmente tale limite è legato alla problematica precedentemente illustrata e alla dimensione dei display all'interno del visore, poiché anche in questo caso perché un gesto possa essere riconosciuto deve essere eseguito nella zona centrale del campo visivo; questo vincolo obbliga l'utente a tenere costantemente il braccio teso davanti a sé, riducendo la naturalezza dei comandi gestuali.

6.3. Costi di sviluppo dell'applicazione

Nella valutazione del costo del progetto va sicuramente incluso il prezzo di acquisto del visore, correntemente di 3000 dollari (circa 2670 euro); i costi richiesti per l'utilizzo dei software sono invece minimi: sia Unity che Blender sono completamente gratuiti e Visual Studio, da acquistare, è considerato dotazione di base di un'azienda di sviluppo software. Per i costi di apprendimento e di configurazione per poter iniziare lo sviluppo è stato valutato un tempo complessivo di 10 giornate. Lo sviluppo vero e proprio dell'applicazione, invece, ha richiesto all'incirca 40 giornate lavorative.

A causa della struttura corrente dell'applicazione si prevedono dei costi di manutenzione piuttosto elevati: al momento, infatti, i modelli presentati all'utente sono in numero limitato ma soprattutto sono statici. Ne consegue che per mantenere l'applicazione allineata con eventuali modifiche del catalogo di vendita sarà necessario intervenire sul codice per inserire i nuovi modelli; analogamente per modifiche al campionario dei colori disponibili. In questa tesi tale difficoltà è stata ignorata, in quanto ci si trova ancora in una fase di

valutazione della possibile diffusione dell'applicazione; nel caso di una reale messa in commercio del software si potrebbero minimizzare i costi di manutenzione modificando la struttura dell'applicazione. Si potrebbe pensare ad un sistema in cui i modelli disponibili non siano statici, ma vengano letti dinamicamente da un database, da aggiornare periodicamente per offrire agli utenti un campionario allineato con la reale offerta di vendita.

Sia i costi di configurazione che quelli di formazione dei futuri utenti sono invece praticamente nulli: una volta installata l'applicazione sul proprio visore HoloLens si può immediatamente iniziare l'utilizzo e durante l'esecuzione all'utente verranno mostrati appositi testi introduttivi che lo guideranno nelle varie operazioni.

6.4. Presentazione dell'applicazione a Farete 2016

“Farete” è un salone annuale promosso da Unindustria Bologna che nasce con l'intenzione di essere un punto d'incontro per le aziende bolognesi. GetConnected prende parte annualmente all'esposizione con un proprio stand e si è presentata la possibilità di partecipare all'edizione del 2016, tenutasi il 5 e il 6 Settembre presso la Fiera di Bologna e che ha contato circa 14.000 presenze e 600 imprese espositrici (Figura 32).

Tale opportunità è stata colta per presentare al pubblico il visore HoloLens e l'applicazione realizzata per Poltronesofà, il cui sviluppo al momento dell'esposizione era nelle ultime fasi di test. È stata un'occasione utile per valutare la risposta del pubblico all'applicazione, per evidenziare eventuali difficoltà di utilizzo e possibili migliorie, ma anche per valutare la reazione dei partecipanti ad una tecnologia innovativa e ancora poco diffusa quale la Mixed Reality. Dal confronto con gli utenti che hanno provato ad utilizzare l'applicazione sono emerse informazioni utili, che sono state utilizzate in seguito per migliorare l'esperienza utente: sono stati individuati i punti cruciali dell'esecuzione, in cui molti degli utenti si sono fermati a chiedere istruzioni perché incerti sull'azione da compiere, ed in tali punti sono state introdotte delle didascalie esplicative per guidare l'utilizzatore. Inoltre, sono state modificate alcune delle simbologie adottate nei menu interattivi, laddove erano risultate ambigue o di difficile interpretazione per gli utenti che si erano trovati ad utilizzarle.



Figura 32 - Foto dell'evento Farete 2016

È stato inoltre interessante osservare la reazione dei partecipanti all'esperienza di realtà mista: come prevedibile, trattandosi per gran parte del pubblico del primo approccio a questo tipo di tecnologia e di dispositivo, inizialmente l'utilizzo è risultato lento e piuttosto incerto. Utilizzare i comandi gestuali richiede un certo livello di pratica, poiché solo con l'esperienza si riesce a capire come muovere esattamente le mani e dove posizionarle perché i gesti vengano riconosciuti dal visore. Nonostante la comprensibile incertezza iniziale, quasi tutti i partecipanti sono riusciti in breve tempo ad assimilare i comandi base e ad utilizzare in autonomia le varie applicazioni disponibili sull'HoloLens.

Grazie all'interazione intuitiva e dinamica consentita dalla tecnologia dell'HoloLens, si è riuscito a suscitare la curiosità e l'interesse anche della parte di pubblico più lontano da questo tipo di dispositivi. Si è registrato che, prevedibilmente, nel pubblico nativo digitale il tempo di apprendimento necessario per iniziare ad usare l'HoloLens in autonomia è decisamente ridotto rispetto agli utenti non nativi, ma ciononostante anche tali utenti sono

risultati interessati all'argomento e desiderosi di sperimentarne l'utilizzo. Un'altra interessante differenza apprezzata tra i vari utenti coinvolti nella dimostrazione è emersa dalle proposte di possibili applicazioni della realtà mista: se i più giovani degli utenti risultavano subito attratti dal campo dell'intrattenimento, altri si sono dimostrati interessati all'utilizzo in campo industriale, ad esempio all'impiego degli ologrammi nella fase di design di un prodotto.

Proprio a causa della versatilità della Mixed Reality, che consente infinite applicazioni, si è notato che l'interesse suscitato negli utenti è notevole, come anche la soddisfazione media nei confronti dell'HoloLens, che, sebbene sia un dispositivo ancora in fase di sviluppo, unico nel suo genere, riesce a garantire un'esperienza di realtà mista pienamente soddisfacente.

CONCLUSIONI E POSSIBILI SVILUPPI

Con il presente lavoro di tesi si è voluto analizzare il concetto di Mixed Reality, tecnologia in fase di sviluppo per la quale l'interesse sta crescendo in maniera esponenziale, esplorandone potenzialità di utilizzo, caratteristiche ed eventuali limiti. Il progetto è stato realizzato utilizzando il visore HoloLens prodotto dalla Microsoft, un dispositivo innovativo e appena messo in commercio, tra i visori più avanzati nel campo della realtà mista. In particolare, grazie alla collaborazione con Poltronasofà, ne è stata sperimentata una possibile applicazione in relazione al design di interni.

L'applicazione realizzata consente agli utenti di visualizzare gli elementi di arredamento sotto forma di ologrammi tridimensionali e di interagire con essi muovendoli nello spazio reale: gli elementi virtuali possono infatti essere spostati, rimossi o ruotati come se si trattasse di oggetti concreti. Sono stati utilizzati i servizi offerti dalla piattaforma Windows Holographic per realizzare l'integrazione tra ologrammi e spazio reale, caratteristica fondamentale della realtà mista, cercando di rendere il più possibile realistica l'esperienza dell'utente. Si è osservato come l'aspetto più interessante ed innovativo della Mixed Reality sia di poter unire l'intuitività e la familiarità dell'interazione alle infinite possibilità di applicazione della virtualizzazione: nel nostro caso, ad esempio, all'utente viene presentata un'esperienza vicina alla quotidianità che gli consente di interagire in maniera immediata e comprensibile, ma allo stesso tempo gli vengono offerti vantaggi come l'accesso immediato a tutti gli elementi del catalogo e la possibilità di personalizzazione degli elementi stessi.

Sono state considerate alcune possibili estensioni dell'applicazione sviluppata, volte a migliorarne l'usabilità ed a raggiungere un grado ancora maggiore di intuitività. Ad esempio, si potrebbe pensare ad una gestione più articolata dei comandi gestuali per realizzare azioni alternative al click, come il trascinamento. In tal modo, sarebbe possibile sviluppare un metodo di posizionamento alternativo, che piuttosto che seguire la direzione dello sguardo segua i movimenti della mano dell'utente nello spazio; o anche, potrebbe essere realizzata una trackbar con un cursore trascinabile, da utilizzare nell'operazione di rotazione in sostituzione degli attuali pulsanti. Per quanto riguarda la scelta dei modelli, attualmente limitata a pochi campioni di esempio, si potrebbe pensare ad una modalità di selezione alternativa, magari realizzata tramite una finestra bidimensionale che costituisca

un vero e proprio catalogo sfogliabile; in tal modo si eviterebbero limitazioni sul numero delle alternative proposte, mantenendo la comodità della selezione.

Gli sviluppi di questo progetto, naturalmente, non sono limitati all'applicazione nel campo del design di interni, ma possono essere orientati verso qualsiasi tipo di utilizzo in cui è prevista l'interazione con un oggetto modellato tridimensionalmente: alcuni esempi potrebbero essere la realizzazione di un supporto alla fase di design di un qualsiasi prodotto industriale, oppure un'applicazione per visualizzare i modelli 3D di articoli acquistabili via internet.

Dall'analisi del feedback ricevuta dai primi utenti, l'applicazione è risultata pienamente soddisfacente: il tempo di apprendimento richiesto dagli utenti per iniziare a utilizzarla in autonomia è risultato essere decisamente breve, anche nel caso di utenti poco esperti o al primo approccio con questo tipo di tecnologia. La soddisfazione degli utenti, ed in generale l'interesse dimostrato dal pubblico, danno adito a previsioni ottimistiche riguardo la diffusione futura di questo tipo di tecnologie, anche in considerazione del calo del prezzo che accompagnerebbe la commercializzazione del visore. Nonostante ci si trovi ancora in una fase di sviluppo e di sperimentazione, non è difficile immaginare una diffusione capillare nei prossimi anni di dispositivi simili al Microsoft HoloLens.

BIBLIOGRAFIA

[Alg15] Alger, M. (2015, 01 06). VR Interface Design Manifesto.

[Alt15] Altman, I. (2015, 04 28). Why Google Glass failed and why Apple Watch could too. Retrieved from Forbes: <http://www.forbes.com/sites/ianaltman/2015/04/28/why-google-glass-failed-and-why-apple-watch-could-too/#4d6b72a158ec>

[Dea16] De Agostini, M. (2016, 08 24). Microsoft HoloLens, com'è fatto all'interno e come funziona. Retrieved from Tom's Hardware: <http://www.tomshw.it/news/microsoft-hololens-com-e-fatto-all-interno-e-come-funziona-79468>

[Dim16] Di Marco, M. (2016, 06 21). Realtà virtuale, i videogiochi non sono il principale interesse di tutti. Retrieved from International Business Times: <http://it.ibtimes.com/realta-virtuale-i-videogiochi-non-sono-il-principale-interesse-degli-utenti-1456447>

[Mil94] Milgram, P. (1994). Augmented Reality: A class of displays on the reality-virtuality continuum.

[Pol13] Polsinelli, P. (2013, 12 5). Design a Game. Retrieved from Why is Unity so popular for videogame development?: <http://designagame.eu/2013/12/unity-popular-videogame-development/>

[Pop16] Popper, B. (2016, 04 18). YouTube introduces live 360 video, the gateway drug to virtual reality. Retrieved from The Verge: <http://www.theverge.com/2016/4/18/11450484/youtube-live-360-degree-video-announced-neal-mohan-interview>

[Pra15] Prasuethsut, L. (2015, 05 4). Meta 2 first impressions: AR feels closer than ever. Retrieved from Wearable: <http://www.wearable.com/ar/meta-2-review>

[Roe16] Roettgers, J. (2016, 07 13). Virtual and Augmented Reality investments up to \$2 billion over last 12 months. Retrieved from Variety: <http://variety.com/2016/digital/news/virtual-augmented-reality-vr-ar-investments-2-billion-dollars-1201813497/>

[Smi16] Smith, G. (2016, 08 11). Pokemon Go Statistics. Retrieved from DMR: <http://expandedramblings.com/index.php/pokemon-go-statistics/>

[Sti13] Stinson, L. (2013, 08 20). New Ikea app places furniture in your home. Retrieved from Wired: <https://www.wired.com/2013/08/a-new-ikea-app-lets-you-place-3d-furniture-in-your-home/>

[Tak12] Takahashi, D. (2012, 11 2). Venture Beat. Retrieved from Game developers, start your Unity 3D engines: <http://venturebeat.com/2012/11/02/game-developers-start-your-unity-3d-engines-interview/>

[You16] YouTube. (2016, 04 18). One step closer to reality: introducing 360-degree live streaming and spatial audio on YouTube. Retrieved from Youtube Official Blog: <https://youtube.googleblog.com/2016/04/one-step-closer-to-reality-introducing.html>